# Table of Contents

NetApp Artificial Intelligence Solutions ................................................. 1  
   AI Converged Infrastructures ................................................... 1  
   Data Pipelines, Data Lakes and Management ................................. 15  
   Use Cases .............................................................................. 157  

NetApp Container Solutions ............................................................... 349  
   NVA-1165: Anthos with NetApp .................................................. 349  
   TR-4919: DevOps with NetApp Astra ......................................... 377  
   NVA-1160: Red Hat OpenShift with NetApp ................................. 398  
   NVA-1166: VMware Tanzu with NetApp ...................................... 565  
   Archived Solutions .................................................................. 602  

Data Migration and Data Protection .................................................... 603  
   Data Migration ....................................................................... 603  
   Data Protection ....................................................................... 687  
   Security .................................................................................. 688  

Enterprise Applications ...................................................................... 689  

NetApp Enterprise Database Solutions ............................................... 690  
   Oracle Database ....................................................................... 690  
   Microsoft SQL Server ............................................................... 1182  
   Open Source Databases ............................................................. 1278  
   SnapCenter for Databases ......................................................... 1288  
   DB Automation Toolkits .............................................................. 1514  

NetApp Modern Data Analytics Solutions .......................................... 1529  
   Cloud Data Management with NetApp File-Object Duality and AWS SageMaker .................................................. 1529  
   Apache Kafka workloads with NetApp NFS storage .................... 1557  
   Confluent Kafka with NetApp ONTAP storage controllers .......... 1604  
   NetApp Storage Solutions for Apache Spark ............................ 1616  
   Big Data Analytics Data to Artificial Intelligence ....................... 1663  
   Best practices for Confluent Kafka ......................................... 1707  
   NetApp hybrid cloud data solutions - Spark and Hadoop based on customer use cases ............................................. 1736  
   Modern data analytics - Different solutions for different analytics strategies ................................................. 1754  
   TR-4623: NetApp E-Series E5700 and Splunk Enterprise ............ 1754  
   NVA-1157-DEPLOY: Apache Spark workload with NetApp storage solution .................................................. 1754  

NetApp Hybrid Multicloud with VMware Solutions .............................. 1755  
   VMware for Public Cloud .......................................................... 1755  
   VMware Hybrid Cloud Use Cases ............................................. 1874  
   NetApp Hybrid Multicloud with VMware Solutions .................... 1877  
   NetApp Hybrid Multicloud with VMware Solutions .................... 2018  
   NetApp Hybrid Multicloud with VMware Solutions .................... 2094  
   BlueXP Backup and Recovery .................................................... 2185  

VMware Sovereign Cloud ................................................................ 2226  
   VMware Resources for Sovereign Cloud .................................. 2226  
   NetApp and VMware Sovereign Cloud ...................................... 2226  

NetApp Hybrid Multicloud with Red Hat OpenShift Container workloads 2228
NetApp Artificial Intelligence Solutions

AI Converged Infrastructures

NetApp AFF A400 with Lenovo ThinkSystem SR670 V2 for AI and ML Model Training

TR-4810: NetApp AFF A400 with Lenovo ThinkSystem SR670 V2 for AI and ML Model Training

Sathish Thyagarajan, David Arnette, NetApp
Mircea Troaca, Lenovo

This solution presents a mid-range cluster architecture using NetApp storage and Lenovo servers optimized for artificial intelligence (AI) workloads. It is meant for small- to medium-sized enterprises for which most compute jobs are single node (single or multi-GPU) or distributed over a few computational nodes. This solution aligns with most day-to-day AI training jobs for many businesses.

This document covers testing and validation of a compute and storage configuration consisting of eight-GPU Lenovo SR670V2 servers, a mid-range NetApp AFF A400 storage system and 100GbE interconnect switch. To measure the performance, we used ResNet50 with the ImageNet dataset, a batch size of 408, half precision, CUDA, and cuDNN. This architecture provides an efficient and cost-effective solution for small and medium-sized organizations just starting out with AI initiatives that require the enterprise-grade capabilities of NetApp ONTAP cloud-connected data storage.

Target audience

This document is intended for the following audiences:

• Data scientists, data engineers, data administrators, and developers of AI systems
• Enterprise architects who design solutions for the development of AI models
• Data scientists and data engineers who are looking for efficient ways to achieve deep learning (DL) and machine learning (ML) development goals
• Business leaders and OT/IT decision makers who want to achieve the fastest possible time to market for AI initiatives

Solution architecture

This solution with Lenovo ThinkSystem servers and NetApp ONTAP with AFF storage is designed to handle AI training on large datasets using the processing power of GPUs alongside traditional CPUs. This validation demonstrates high performance and optimal data management with a scale-out architecture that uses either one, two, or four Lenovo SR670 V2 servers alongside a single NetApp AFF A400 storage system. The following figure provides an architectural overview.

This NetApp and Lenovo solution offers the following key benefits:

• Highly efficient and cost-effective performance when executing multiple training jobs in parallel
• Scalable performance based on different numbers of Lenovo servers and different models of NetApp
storage controllers

- Robust data protection to meet low recovery point objectives (RPOs) and recovery time objectives (RTOs) with no data loss
- Optimized data management with snapshots and clones to streamline development workflows

Technology overview

This section introduces the major components of this solution in greater detail.

NetApp AFF systems

NetApp AFF storage systems enable businesses to meet enterprise storage requirements with industry-leading performance, superior flexibility, cloud integration, and best-in-class data management. Designed specifically for flash, AFF systems help accelerate, manage, and protect business-critical data.

NetApp AFF A400 is a mid-range NVMe flash storage system that includes the following features:

- Maximum effective capacity: ~20PB
- Maximum scale-out: 2-24 nodes (12 HA pairs)
- 25GbE and 16Gb FC host support
- 100GbE RDMA over Converged Ethernet (RoCE) connectivity to NVMe expansion storage shelves
- 100GbE RoCE ports can be used for host network attachment if NVMe shelves aren’t attached
- Full 12Gbps SAS connectivity expansion storage shelves
- Available in two configurations:
  - Ethernet: 4x 25Gb Ethernet (SFP28) ports
  - Fiber Channel: 4x 16Gb FC (SFP+) ports
- 100% 8KB random read @.4 ms 400k IOPS

NetApp AFF A250 features for entry level AI/ML deployments include the following:

- Maximum effective capacity: 35PB
• Maximum scale out: 2-24 nodes (12 HA pairs)
• 440k IOPS random reads @1ms
• Built on the latest NetApp ONTAP release ONTAP 9.8 or later
• Two 25Gb Ethernet ports for HA and cluster interconnect

NetApp also offers other storage systems, such as the AFF A800 and AFF A700 that provide higher performance and scalability for larger-scale AI/ML deployments.

NetApp ONTAP

ONTAP 9, the latest generation of storage management software from NetApp, enables businesses to modernize infrastructure and transition to a cloud-ready data center. Leveraging industry-leading data management capabilities, ONTAP enables the management and protection of data with a single set of tools, regardless of where that data resides. Data can also be moved freely to wherever it’s needed: the edge, the core, or the cloud. ONTAP 9 includes numerous features that simplify data management, accelerate and protect critical data, and future-proof infrastructure across hybrid cloud architectures.

Simplify data management

Data management is crucial to enterprise IT operations so that appropriate resources are used for applications and datasets. ONTAP includes the following features to streamline and simplify operations and reduce the total cost of operation:

• **Inline data compaction and expanded deduplication.** Data compaction reduces wasted space inside storage blocks, and deduplication significantly increases effective capacity. This applies to data stored locally and data tiered to the cloud.

• **Minimum, maximum, and adaptive quality of service (QoS).** Granular QoS controls help maintain performance levels for critical applications in highly shared environments.

• **ONTAP FabricPool.** This feature automatically tiers cold data to public and private cloud storage options, including Amazon Web Services (AWS), Azure, and NetApp StorageGRID object storage.

Accelerate and protect data

ONTAP delivers superior levels of performance and data protection and extends these capabilities in the following ways:

• **Performance and lower latency.** ONTAP offers the highest possible throughput at the lowest possible latency.

• **Data protection.** ONTAP provides built-in data protection capabilities with common management across all platforms.

• **NetApp Volume Encryption.** ONTAP offers native volume-level encryption with both onboard and external key management support.

Future-proof infrastructure

ONTAP 9 helps meet demanding and constantly changing business needs:

• **Seamless scaling and nondisruptive operations.** ONTAP supports the nondisruptive addition of capacity to existing controllers as well as to scale-out clusters. Customers can upgrade to the latest technologies, such as NVMe and 32Gb FC, without costly data migrations or outages.

• **Cloud connection.** ONTAP is the most cloud-connected storage management software, with options for
software-defined storage (ONTAP Select) and cloud-native instances (NetApp Cloud Volumes Service) in all public clouds.

- **Integration with emerging applications.** ONTAP offers enterprise-grade data services for next-generation platforms and applications such as OpenStack, Hadoop, and MongoDB by using the same infrastructure that supports existing enterprise apps.

**NetApp FlexGroup volumes**

Training datasets are typically a collection of potentially billions of files. Files can include text, audio, video, and other forms of unstructured data that must be stored and processed to be read in parallel. The storage system must store many small files and must read those files in parallel for sequential and random I/O.

A FlexGroup volume (the following figure) is a single namespace made up of multiple constituent member volumes that is managed and acts like a NetApp FlexVol volume to storage administrators. Files in a FlexGroup volume are allocated to individual member volumes and are not striped across volumes or nodes. They enable the following capabilities:

- Up to 20 petabytes of capacity and predictable low latency for high-metadata workloads
- Up to 400 billion files in the same namespace
- Parallelized operations in NAS workloads across CPUs, nodes, aggregates, and constituent FlexVol volumes

![NetApp FlexGroup volume diagram](image)

**Lenovo ThinkSystem portfolio**

Lenovo ThinkSystem servers feature innovative hardware, software, and services that solve customers' challenges today and deliver an evolutionary, fit-for-purpose, modular design approach to address tomorrow's challenges. These servers capitalize on best-in-class, industry-standard technologies coupled with differentiated Lenovo innovations to provide the greatest possible flexibility in x86 servers.

Key advantages of deploying Lenovo ThinkSystem servers include the following:
• Highly scalable, modular designs that grow with your business
• Industry-leading resilience to save hours of costly unscheduled downtime
• Fast flash technologies for lower latencies, quicker response times, and smarter data management in real time

In the AI area, Lenovo is taking a practical approach to helping enterprises understand and adopt the benefits of ML and AI for their workloads. Lenovo customers can explore and evaluate Lenovo AI offerings in Lenovo AI Innovation Centers to fully understand the value for their particular use case. To improve time to value, this customer-centric approach gives customers proofs of concept for solution development platforms that are ready to use and optimized for AI.

**Lenovo SR670 V2**

The Lenovo ThinkSystem SR670 V2 rack server delivers optimal performance for accelerated AI and high-performance computing (HPC). Supporting up to eight GPUs, the SR670 V2 is suited for the computationally intensive workload requirements of ML, DL, and inference.
With the latest scalable Intel Xeon CPUs that support high-end GPUs (including the NVIDIA A100 80GB PCIe 8x GPU), the ThinkSystem SR670 V2 delivers optimized, accelerated performance for AI and HPC workloads.

Because more workloads use the performance of accelerators, the demand for GPU density has increased. Industries such as retail, financial services, energy, and healthcare are using GPUs to extract greater insights and drive innovation with ML, DL, and inference techniques.

The ThinkSystem SR670 V2 is an optimized, enterprise-grade solution for deploying accelerated HPC and AI workloads in production, maximizing system performance while maintaining data center density for supercomputing clusters with next-generation platforms.

Other features include:

- Support for GPU direct RDMA I/O in which high-speed network adapters are directly connected to the GPUs to maximize I/O performance.
• Support for GPU direct storage in which NVMe drives are directly connected to the GPUs to maximize storage performance.

MLPerf

MLPerf is the industry-leading benchmark suite for evaluating AI performance. In this validation, we used its image-classification benchmark with MXNet, one of the most popular AI frameworks. The MXNet_benchmarks training script was used to drive AI training. The script contains implementations of several popular conventional models and is designed to be as fast as possible. It can be run on a single machine or run in distributed mode across multiple hosts.

Test plan

In this validation, we performed image recognition training as specified by MLPerf v2.0. Specifically, we trained the ResNet v2.0 model with the ImageNet dataset until we reached an accuracy of 76.1%. The main metric is the time to reach the desired accuracy. We also report training bandwidth in images per second to better judge scale-out efficiency.

The primary test case evaluated multiple independent training processes (one per node) running concurrently. This simulates the main use case, a shared system used by multiple data scientists. The second test case evaluated scale-out efficiency.

Test results

The following table summarizes the results for all tests performed for this solution.

<table>
<thead>
<tr>
<th>Test description</th>
<th>Results summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image recognition training: multiple concurrent jobs</td>
<td>Highly efficient performance. All jobs ran at full speed even when the cluster was fully used. The NetApp storage systems delivered training performance comparable to local SSD storage while enabling easy sharing of data between servers.</td>
</tr>
<tr>
<td>Image recognition training: scale out</td>
<td>Highly efficient for up to four nodes. At that point, scale out was less efficient but still feasible. Using a higher-speed computational network improves scalability. The NetApp storage system delivered training performance comparable to local SSD storage while enabling easy sharing of data between servers.</td>
</tr>
</tbody>
</table>

Test configuration

This section describes the tested configurations, the network infrastructure, the SR670 V2 server, and the NetApp storage provisioning details.

Solution architecture

We used the solution components listed in the following table for this validation.
In this validation, we used ResNet v2.0 with the ImageNet basis set as specified by MLPerf v2.0. The dataset is stored in a NetApp AFF storage system with the NFS protocol. The SR670s were connected to the NetApp AFF A400 storage system over a 100GbE switch.

ImageNet is a frequently used image dataset. It contains almost 1.3 million images for a total size of 144GB. The average image size is 108KB.

The following figure depicts the network topology of the tested configuration.
The /a400-100g folder contains the dataset used for ResNet validation.

**Test procedure and detailed results**

This section describes the detailed test procedure results.

**Image recognition training using ResNet in ONTAP**

We ran the ResNet50 benchmark with one and two SR670 V2 servers. This test used the MXNet 22.04-py3 NGC container to run the training.

We used the following test procedure in this validation:

1. We cleared the host cache before running the script to make sure that data was not already cached:

   ```bash
   sync ; sudo /sbin/sysctl vm.drop_caches=3
   ```

2. We ran the benchmark script with the ImageNet dataset in server storage (local SSD storage) as well as on the NetApp AFF storage system.

3. We validated network and local storage performance using the `dd` command.

4. For the single-node run, we used the following command:
5. For the distributed runs, we used the parameter server’s parallelization model. We used two parameter servers per node, and we set the number of epochs to be the same as for the single-node run. We did this because distributed training often takes more epochs due to imperfect synchronization between processes. The different number of epochs can skew comparisons between single-node and distributed cases.

Data read speed: Local versus network storage

The read speed was tested by using the `dd` command on one of the files for the ImageNet dataset. Specifically, we ran the following commands for both local and network data:

```
sync ; sudo /sbin/sysctl vm.drop_caches=3dd if=/a400-100g/netapp-ra/resnet/data/preprocessed_data/train.rec of=/dev/null bs=512k count=2048
```

Results (average of 5 runs):

- Local storage: 1.7 GB/s
- Network storage: 1.5 GB/s.

Both values are similar, demonstrating that the network storage can deliver data at a rate similar to local storage.

Shared use case: Multiple, independent, simultaneous jobs

This test simulated the expected use case for this solution: multi-job, multi-user AI training. Each node ran its own training while using the shared network storage. The results are displayed in the following figure, which shows that the solution case provided excellent performance with all jobs running at essentially the same speed as individual jobs. The total throughput scaled linearly with the number of nodes.
These graphs present the runtime in minutes and the aggregate images per second for compute nodes that used eight GPUs from each server on 100 GbE client networking, combining both the concurrent training model and the single training model. The average runtime for the training model was 35 minutes and 9 seconds. The individual runtimes were 34 minutes and 32 seconds, 36 minutes and 21 seconds, 34 minutes and 37 seconds, 35 minutes and 25 seconds, and 34 minutes and 31 seconds. The average images per second for the training model were 22,573, and the individual images per second were 21,764; 23,438; 22,556; 22,564; and 22,547.

Based on our validation, one independent training model with a NetApp data runtime was 34 minutes and 54 seconds with 22,231 images/sec. One independent training model with a local data (DAS) runtime was 34 minutes and 21 seconds with 22,102 images/sec. During those runs the average GPU utilization was 96%, as
observed on nvidia-smi. Note that this average includes the testing phase, during which GPUs were not used, while CPU utilization was 40% as measured by mpstat. This demonstrates that the data delivery rate is sufficient in each case.

**Architecture adjustments**

The setup used for this validation can be adjusted to fit other use cases.

**CPU Adjustments**

We used a Skylake Intel Xeon Platinum 8360Y processor for this validation, as recommended by Lenovo. We expect that the equivalent Cascade Lake CPU, an Intel Xeon Gold 6330 processor, would deliver similar performance because this workload is not CPU bound.

**Storage Capacity Increase**

Based on your storage capacity needs, you can increase the share storage (NFS volume) on demand, provided that you have the additional disk shelves and controller models. You can do this from the CLI or from the NetApp web interface of the storage controller as the admin user.

**Conclusion**

The NetApp and Lenovo solution validated here is a flexible scale-out architecture that is ideal for entry into mid-level enterprise AI.

NetApp storage delivers the same or better performance as local SSD storage and offers the following benefits to data scientists, data engineers, and IT decision makers:

- Effortless sharing of data between AI systems, analytics, and other critical business systems. This data sharing reduces infrastructure overhead, improves performance, and streamlines data management across the enterprise.
- Independently scalable compute and storage to minimize costs and improve resource utilization.
- Streamlined development and deployment workflows using integrated snapshots and clones for instantaneous and space-efficient user workspaces, integrated version control, and automated deployment.
- Enterprise-grade data protection for disaster recovery and business continuance.

**Acknowledgments**

- Karthikeyan Nagalingam, Technical Marketing Engineer, NetApp
- Jarrett Upton, Admin, AI Lab Systems, Lenovo

**Where to find additional information**

To learn more about the information described in this document, refer to the following documents and/or websites:

- NetApp All Flash Arrays product page
  

- NetApp AFF A400 page
  
  https://docs.netapp.com/us-en/ontap-systems/a400/index.html
NetApp ONTAP data management software product page

MLPerf
https://mlperf.org

TensorFlow benchmark
https://github.com/tensorflow/benchmarks

NVIDIA SMI (nvidia-smi)

NetApp AI with NVIDIA
Overview of ONTAP AI converged infrastructure solutions from NetApp and NVIDIA.

NetApp ONTAP AI with NVIDIA DGX A100 Systems

- Design Guide
- Deployment Guide

NetApp ONTAP AI with NVIDIA DGX A100 Systems and Mellanox Spectrum Ethernet Switches

- Design Guide
- Deployment Guide

NVA-1151-DESIGN: NetApp ONTAP AI with NVIDIA DGX A100 systems design guide
David Arnette and Sung-Han Lin, NetApp

NVA-1151-DESIGN describes a NetApp Verified Architecture for machine learning and artificial intelligence workloads using NetApp AFF A800 storage systems, NVIDIA DGX A100 systems, and NVIDIA Mellanox network switches. It also includes benchmark test results for the architecture as implemented.

NVA-1151-DEPLOY: NetApp ONTAP AI with NVIDIA DGX A100 systems
David Arnette, NetApp

NVA-1151-DEPLOY includes storage system deployment instructions for a NetApp Verified Architecture (NVA) for machine learning (ML) and artificial intelligence (AI) workloads using NetApp AFF A800 storage systems, NVIDIA DGX A100 systems, and NVIDIA Mellanox network switches. It also includes instructions for running validation benchmark tests after deployment is complete.
NVA-1151-DEPLOY: NetApp ONTAP AI with NVIDIA DGX A100 systems

NVA-1153-DESIGN: NetApp ONTAP AI with NVIDIA DGX A100 systems and Mellanox Spectrum Ethernet switches

David Arnette and Sung-Han Lin, NetApp

NVA-1153-DESIGN describes a NetApp Verified Architecture for machine learning (ML) and artificial intelligence (AI) workloads using NetApp AFF A800 storage systems, NVIDIA DGX A100 systems, and NVIDIA Mellanox Spectrum SN3700V 200Gb Ethernet switches. This design features RDMA over Converged Ethernet (RoCE) for the compute cluster interconnect fabric to provide customers with a completely ethernet-based architecture for high-performance workloads. This document also includes benchmark test results for the architecture as implemented.

NVA-1153-DEPLOY: NetApp ONTAP AI with NVIDIA DGX A100 systems and Mellanox Spectrum Ethernet switches

NVA-1153-DEPLOY includes storage-system deployment instructions for a NetApp Verified Architecture for machine learning (ML) and artificial intelligence (AI) workloads using NetApp AFF A800 storage systems, NVIDIA DGX A100 systems, and NVIDIA Mellanox Spectrum SN3700V 200Gb Ethernet switches. It also includes instructions for executing validation benchmark tests after deployment is complete.

NetApp EF-Series AI with NVIDIA

Overview of EF-Series AI converged infrastructure solutions from NetApp and NVIDIA.

EF-Series AI with NVIDIA DGX A100 Systems and BeeGFS

• Design Guide
• Deployment Guide
• BeeGFS Deployment Guide

NVA-1156-DESIGN: NetApp EF-Series AI with NVIDIA DGX A100 systems and BeeGFS

Abdel Sadek, Tim Chau, Joe McCormick and David Arnette, NetApp

NVA-1156-DESIGN describes a NetApp Verified Architecture for machine learning (ML) and artificial intelligence (AI) workloads using NetApp EF600 NVMe storage systems, the BeeGFS parallel file system, NVIDIA DGX A100 systems, and NVIDIA Mellanox Quantum QM8700 200Gbps IB switches. This design features 200Gbps InfiniBand (IB)
for the storage and compute cluster interconnect fabric to provide customers with a completely IB-based architecture for high-performance workloads. This document also includes benchmark test results for the architecture as implemented.

NVA-1156-DESIGN: NetApp EF-Series AI with NVIDIA DGX A100 systems and BeeGFS

NVA-1156-DEPLOY: NetApp EF-Series AI with NVIDIA DGX A100 systems and BeeGFS

Abdel Sadek, Tim Chau, Joe McCormick, and David Arnette, NetApp

This document describes a NetApp Verified Architecture for machine learning (ML) and artificial intelligence (AI) workloads using NetApp EF600 NVMe storage systems, the ThinkParQ BeeGFS parallel file system, NVIDIA DGX A100 systems, and NVIDIA Mellanox Quantum QM8700 200Gbps InfiniBand (IB) switches. This document also includes instructions for executing validation benchmark tests after the deployment is complete.

NVA-1156-DEPLOY: NetApp EF-Series AI with NVIDIA DGX A100 systems and BeeGFS

TR-4859: Deploying IBM spectrum scale with NetApp E-Series storage - Installation and validation

Chris Seirer, NetApp

TR-4859 describes the process of deploying a full parallel file system solution based on IBM's Spectrum Scale software stack. TR-4859 is designed to provide details on how to install Spectrum Scale, validate the infrastructure, and manage the configuration.

TR-4859: Deploying IBM spectrum scale with NetApp E-Series storage - Installation and validation

TR-4815: NetApp AFF A800 and Fujitsu Server PRIMERGY GX2570 M5 for AI and ML model training workloads

David Arnette, NetApp
Takashi Oishi, Fujitsu

This solution focuses on a scale-out architecture to deploy artificial intelligence systems with NetApp storage systems and Fujitsu servers. The solution was validated with MLperf v0.6 model-training benchmarks using Fujitsu GX2570 servers and a NetApp AFF A800 storage system.

TR-4815: NetApp AFF A800 and Fujitsu Server PRIMERGY GX2570 M5 for AI and ML model training workloads

Data Pipelines, Data Lakes and Management

AWS FSx for NetApp ONTAP (FSxN) for MLOps

Author(s):
Jian Jian (Ken), Senior Data & Applied Scientist, NetApp

This section delves into the practical application of AI infrastructure development, providing an end-to-end walkthrough of constructing an MLOps pipeline using FSxN. Comprising three comprehensive examples, it guides you to meet your MLOps needs via this powerful data management platform.

These articles focus on:

1. Part 1 - Integrating AWS FSx for NetApp ONTAP (FSxN) as a private S3 bucket into AWS SageMaker
2. Part 2 - Leveraging AWS FSx for NetApp ONTAP (FSxN) as a Data Source for Model Training in SageMaker
3. Part 3 - Building A Simplified MLOps Pipeline (CI/CT/CD)

By the end of this section, you will have gained a solid understanding of how to use FSxN to streamline MLOps processes.

Part 1 - Integrating AWS FSx for NetApp ONTAP (FSxN) as a private S3 bucket into AWS SageMaker

Author(s):
Jian Jian (Ken), Senior Data & Applied Scientist, NetApp

Introduction
Using SageMaker as an example, this page provides guidance on configuring FSxN as a private S3 bucket.

For more information about FSxN, please take a look at this presentation (Video Link)

User Guide

Server creation

Create a SageMaker Notebook Instance

1. Open AWS console. In the search panel, search SageMaker and click the service Amazon SageMaker.

2. Open the Notebook instances under Notebook tab, click the orange button Create notebook instance.
3. In the creation page,
   Enter the **Notebook instance name**
   Expand the **Network** panel
   Leave other entries default and select a **VPC, Subnet**, and **Security group(s)**. (This **VPC** and **Subnet** will be used to create FSxN file system later)
   Click the orange button **Create notebook instance** at the bottom right.
Create notebook instance

Amazon SageMaker provides pre-built fully managed notebook instances that run Jupyter notebooks. The notebook instances include example code for common model training and hosting exercises. Learn more

Notebook instance settings

- **Notebook instance name**: fxm-demo
  - Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.
- **Notebook instance type**: ml.t3.medium
- **Elastic Inference**: Learn more
  - None
- **Platform identifier**: Learn more
  - Amazon Linux 2, Jupyter Lab

Permissions and encryption

- **IAM role**: AmazonSageMakerServiceCatalogProductsUseRole
  - Create role using the role creation wizard
- **Root access - optional**
  - Enable - Give users root access to the notebook
  - Disable - Don’t give users root access to the notebook
  - Lifecycle configurations always have root access
- **Encryption key - optional**
  - Encrypt your notebook data. Choose an existing KMS key or enter a key's ARN

Network - optional

- **VPC - optional**
  - Default vpc-0df3966ab1fca2ec9 (172.31.0.0/16)
- **Subnet**
  - subnet-0060df0d0f562672 (172.31.16.0/20) | us-east-1a
- **Security group(s)**

Git repositories - optional

Tags - optional

Create notebook instance
Create an FSxN File System

1. Open AWS console. In the search panel, search Fsx and click the service FSx.

2. Click **Create file system**.

3. Select the first card **FSx for NetApp ONTAP** and click **Next**.
4. In the details configuration page.
   a. Select the **Standard create** option.

   b. Enter the **File system name** and the **SSD storage capacity**.
c. Make sure to use the **VPC** and **subnet** same to the **SageMaker Notebook** instance.
d. Enter the **Storage virtual machine** name and **Specify a password** for your SVM (storage virtual machine).
e. Leave other entries default and click the orange button **Next** at the bottom right.

f. Click the orange button **Create file system** at the bottom right of the review page.
5. It may takes about **20-40 minutes** to spin up the FSx file system.

**Server Configuration**

**ONTAP Configuration**

1. Open the created FSx file system. Please make sure the status is **Available**.

2. Select the **Administration** tab and keep the **Management endpoint - IP address** and **ONTAP administrator username**.
3. Open the created **SageMaker Notebook instance** and click **Open JupyterLab**.

4. In the Jupyter Lab page, open a new **Terminal**.
5. Enter the ssh command `ssh <admin user name>@<ONTAP server IP>` to login to the FSxN ONTAP file system. (The user name and IP address are retrieved from the step 2) Please use the password used when creating the **Storage virtual machine**.

6. Execute the commands in the following order.
   We use `fsxn-ontap` as the name for the **FSxN private S3 bucket name**. Please use the **storage virtual machine name** for the `-vserver` argument.

   ```bash
   vserver object-store-server create -vserver fsxn-svm-demo -object-store-server fsx_s3 -is-http-enabled true -is-https-enabled false
   vserver object-store-server user create -vserver fsxn-svm-demo -user s3user
   vserver object-store-server group create -name s3group -users s3user -policies FullAccess
   vserver object-store-server bucket create fsxn-ontap -vserver fsxn-svm-demo -type nas -nas-path /vol1
   ```
7. Execute the below commands to retrieve the endpoint IP and credentials for FSxN private S3.

```
(network interface show -vserver fsxn-svm-demo -lif nfs_smb_management_1
set adv
vserver object-store-server user show
```

8. Keep the endpoint IP and credential for future use.
Client Configuration

1. In SageMaker Notebook instance, create a new Jupyter notebook.

2. Use the below code as a work around solution to upload files to FSxN private S3 bucket. For a comprehensive code example please refer to this notebook. fsxn_demo.ipynb

```python
# Setup configurations
# ------- Manual configurations -------
seed: int = 77  # Random
seed
bucket_name: str = 'fsxn-ontap'  # The bucket name in ONTAP
aws_access_key_id = '<Your ONTAP bucket key id>'  # Please get this credential from ONTAP
aws_secret_access_key = '<Your ONTAP bucket access key>'  # Please get this credential from ONTAP
fsx_endpoint_ip: str = '<Your FSxN IP address>'  # Please get this IP address from FSXN
# ------- Manual configurations -------

# Workaround
## Permission patch
mkdir -p vol1
sudo mount -t nfs $fsx_endpoint_ip:/vol1 /home/ec2-user/SageMaker/vol1
sudo chmod 777 /home/ec2-user/SageMaker/vol1

## Authentication for FSxN as a Private S3 Bucket
aws configure set aws_access_key_id $aws_access_key_id
```
aws configure set aws_secret_access_key $aws_secret_access_key

## Upload file to the FSxN Private S3 Bucket

```bash
aws s3 cp --endpoint-url http://\$fsx_endpoint_ip /home/ec2-user
/SageMaker/\$local_file_path  s3://\$bucket_name/\$local_file_path
```

# Read data from FSxN Private S3 bucket

## Initialize a s3 resource client

```python
import boto3

# Get session info
region_name = boto3.session.Session().region_name

# Initialize Fsxn S3 bucket object
# --- Start integrating SageMaker with FSXN ---
# This is the only code change we need to incorporate SageMaker with FSXN
s3_client: boto3.client = boto3.resource('s3',
    region_name=region_name,
    aws_access_key_id=aws_access_key_id,
    aws_secret_access_key=aws_secret_access_key,
    use_ssl=False,
    endpoint_url=f'http://\$fsx_endpoint_ip',
    config=boto3.session.Config(
        signature_version='s3v4',
        s3={'addressing_style': 'path'}
    )
)
# --- End integrating SageMaker with FSXN ---

## Read file byte content

```python
bucket = s3_client.Bucket(bucket_name)

binary_data = bucket.Object(data.filename).get()]['Body']
```

This concludes the integration between FSxN and the SageMaker instance.

**Useful debugging checklist**

- Ensure that the SageMaker Notebook instance and FSxN file system are in the same VPC.
- Remember to run the **set dev** command on ONTAP to set the privilege level to dev.
FAQ (As of Sep 27, 2023)

Q: Why am I getting the error "An error occurred (NotImplemented) when calling the CreateMultipartUpload operation: The s3 command you requested is not implemented" when uploading files to FSxN?

A: As a private S3 bucket, FSxN supports uploading files up to 100MB. When using the S3 protocol, files larger than 100MB are divided into 100MB chunks, and the 'CreateMultipartUpload' function is called. However, the current implementation of FSxN private S3 does not support this function.

Q: Why am I getting the error "An error occurred (AccessDenied) when calling the PutObject operations: Access Denied" when uploading files to FSxN?

A: To access the FSxN private S3 bucket from a SageMaker Notebook instance, switch the AWS credentials to the FSxN credentials. However, granting write permission to the instance requires a workaround solution that involves mounting the bucket and running the 'chmod' shell command to change the permissions.

Q: How can I integrate the FSxN private S3 bucket with other SageMaker ML services?

A: Unfortunately, the SageMaker services SDK does not provide a way to specify the endpoint for the private S3 bucket. As a result, FSxN S3 is not compatible with SageMaker services such as Sagemaker Data Wrangler, Sagemaker Clarify, Sagemaker Glue, Sagemaker Athena, Sagemaker AutoML, and others.

Part 2 - Leveraging AWS FSx for NetApp ONTAP (FSxN) as a Data Source for Model Training in SageMaker

Author(s):
Jian Jian (Ken), Senior Data & Applied Scientist, NetApp

Introduction

This tutorial offers a practical example of a computer vision classification project, providing hands-on experience in building ML models that utilize FSxN as the data source within the SageMaker environment. The project focuses on using PyTorch, a deep learning framework, to classify tire quality based on tire images. It emphasizes the development of machine learning models using FSxN as the data source in Amazon SageMaker.

What is FSxN

Amazon FSx for NetApp ONTAP is indeed a fully managed storage solution offered by AWS. It leverages NetApp’s ONTAP file system to provide reliable and high-performance storage. With support for protocols like NFS, SMB, and iSCSI, it allows seamless access from different compute instances and containers. The service is designed to deliver exceptional performance, ensuring fast and efficient data operations. It also offers high availability and durability, ensuring that your data remains accessible and protected. Additionally, the storage capacity of Amazon FSx for NetApp ONTAP is scalable, allowing you to easily adjust it according to your needs.

Prerequisite

Network Environment
FSxN (Amazon FSx for NetApp ONTAP) is an AWS storage service. It includes a file system running on the NetApp ONTAP system and an AWS-managed system virtual machine (SVM) that connects to it. In the provided diagram, the NetApp ONTAP server managed by AWS is located outside the VPC. The SVM serves as the intermediary between SageMaker and the NetApp ONTAP system, receiving operation requests from SageMaker and forwarding them to the underlying storage. To access FSxN, SageMaker must be placed within the same VPC as the FSxN deployment. This configuration ensures communication and data access between SageMaker and FSxN.

**Data Access**

In real-world scenarios, data scientists typically utilize the existing data stored in FSxN to build their machine learning models. However, for demonstration purposes, since the FSxN file system is initially empty after creation, it is necessary to manually upload the training data. This can be achieved by mounting FSxN as a volume to SageMaker. Once the file system is successfully mounted, you can upload your dataset to the mounted location, making it accessible for training your models within the SageMaker environment. This approach allows you to leverage the storage capacity and capabilities of FSxN while working with SageMaker for model development and training.

The data reading process involves configuring FSxN as a private S3 bucket. To learn the detailed configuration instructions, please refer to Part 1 - Integrating AWS FSx for NetApp ONTAP (FSxN) as a private S3 bucket into AWS SageMaker

**Integration Overview**
The workflow of using training data in FSxN to build a deep learning model in SageMaker can be summarized into three main steps: data loader definition, model training, and deployment. At a high level, these steps form the foundation of an MLOps pipeline. However, each step involves several detailed sub-steps for a comprehensive implementation. These sub-steps encompass various tasks such as data preprocessing, dataset splitting, model configuration, hyperparameter tuning, model evaluation, and model deployment. These steps ensure a thorough and effective process for building and deploying deep learning models using training data from FSxN within the SageMaker environment.

**Step-by-Step Integration**

**Data Loader**

In order to train a PyTorch deep learning network with data, a data loader is created to facilitate the feeding of data. The data loader not only defines the batch size but also determines the procedure for reading and preprocessing each record within the batch. By configuring the data loader, we can handle the processing of data in batches, enabling training of the deep learning network.

The data loader consists of 3 parts.

**Preprocessing Function**

```python
from torchvision import transforms

preprocess = transforms.Compose([  
    transforms.ToTensor(),
    transforms.Resize((224, 224)),
    transforms.Normalize(  
        mean=[0.485, 0.456, 0.406],
        std=[0.229, 0.224, 0.225]
    )
])
```

The above code snippet demonstrates the definition of image preprocessing transformations using the `torchvision.transforms` module. In this tutorial, the preprocess object is created to apply a series of transformations. Firstly, the `ToTensor()` transformation converts the image into a tensor representation. Subsequently, the `Resize 224,224` transformation resizes the image to a fixed size of 224x224 pixels. Finally, the `Normalize()` transformation normalizes the tensor values by subtracting the mean and dividing by the standard deviation along each channel. The mean and standard deviation values used for normalization are
commonly employed in pre-trained neural network models. Overall, this code prepares the image data for further processing or input into a pre-trained model by converting it to a tensor, resizing it, and normalizing the pixel values.

**The PyTorch Dataset Class**

```python
import torch
from io import BytesIO
from PIL import Image

class FSxNImageDataset(torch.utils.data.Dataset):
    def __init__(self, bucket, prefix='', preprocess=None):
        self.image_keys = [
            s3_obj.key
            for s3_obj in list(bucket.objects.filter(Prefix=prefix).all())
        ]
        self.preprocess = preprocess

    def __len__(self):
        return len(self.image_keys)

    def __getitem__(self, index):
        key = self.image_keys[index]
        response = bucket.Object(key)

        label = 1 if key[13:].startswith('defective') else 0

        image_bytes = response.get()]['Body'].read()
        image = Image.open(BytesIO(image_bytes))
        if image.mode == 'L':
            image = image.convert('RGB')

        if self.preprocess is not None:
            image = self.preprocess(image)

        return image, label
```

This class provides functionality to obtain the total number of records in the dataset and defines the method for reading data for each record. Within the `__getitem__` function, the code utilizes the boto3 S3 bucket object to retrieve the binary data from FSxN. The code style for accessing data from FSxN is similar to reading data from Amazon S3. The subsequent explanation delves into the creation process of the private S3 object bucket.

**FSxN as a private S3 repository**
import boto3

# Get session info
region_name = boto3.session.Session().region_name

# Initialize Fsxn S3 bucket object
# --- Start integrating SageMaker with FSXN ---
# This is the only code change we need to incorporate SageMaker with FSXN
s3_client: boto3.client = boto3.resource('s3',
             region_name=region_name,
             aws_access_key_id=aws_access_key_id,
             aws_secret_access_key=aws_secret_access_key,
             use_ssl=False,
             endpoint_url=f'http://{fsx_endpoint_ip}',
             config=boto3.session.Config(
                 signature_version='s3v4',
                 s3={'addressing_style': 'path'}
             )
)
# s3_client = boto3.resource('s3')
bucket = s3_client.Bucket(bucket_name)
# --- End integrating SageMaker with FSXN ---

To read data from FSxN in SageMaker, a handler is created that points to the FSxN storage using the S3 protocol. This allows FSxN to be treated as a private S3 bucket. The handler configuration includes specifying the IP address of the FSxN SVM, the bucket name, and the necessary credentials. For a comprehensive explanation on obtaining these configuration items, please refer to the document at Part 1 - Integrating AWS FSx for NetApp ONTAP (FSxN) as a private S3 bucket into AWS SageMaker.

In the example mentioned above, the bucket object is used to instantiate the PyTorch dataset object. The dataset object will be further explained in the subsequent section.
The PyTorch Data Loader

```python
from torch.utils.data import DataLoader
torch.manual_seed(seed)

# 1. Hyperparameters
batch_size = 64

# 2. Preparing for the dataset
dataset = FSxNImageDataset(bucket, 'dataset/tyre', preprocess=preprocess)

train, test = torch.utils.data.random_split(dataset, [1500, 356])
data_loader = DataLoader(dataset, batch_size=batch_size, shuffle=True)
```

In the example provided, a batch size of 64 is specified, indicating that each batch will contain 64 records. By combining the PyTorch Dataset class, the preprocessing function, and the training batch size, we obtain the data loader for training. This data loader facilitates the process of iterating through the dataset in batches during the training phase.

Model Training

```python
from torch import nn

class TyreQualityClassifier(nn.Module):
    def __init__(self):
        super().__init__()
        self.model = nn.Sequential(
            nn.Conv2d(3, 32, (3, 3)),
            nn.ReLU(),
            nn.Conv2d(32, 32, (3, 3)),
            nn.ReLU(),
            nn.Conv2d(32, 64, (3, 3)),
            nn.ReLU(),
            nn.Flatten(),
            nn.Linear(64*(224-6)*(224-6), 2)
        )
    def forward(self, x):
        return self.model(x)
```
import datetime

num_epochs = 2
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

model = TyreQualityClassifier()
fn_loss = torch.nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)

model.to(device)
for epoch in range(num_epochs):
    for idx, (X, y) in enumerate(data_loader):
        X = X.to(device)
        y = y.to(device)

        y_hat = model(X)

        loss = fn_loss(y_hat, y)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        current_time = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        print(f"Current Time: {current_time} - Epoch [{epoch+1}]/{num_epochs}] - Batch [{idx + 1}] - Loss: {loss}", end='\')

This code implements a standard PyTorch training process. It defines a neural network model called TyreQualityClassifier using convolutional layers and a linear layer to classify tire quality. The training loop iterates over data batches, computes the loss, and updates the model’s parameters using backpropagation and optimization. Additionally, it prints the current time, epoch, batch, and loss for monitoring purposes.

Model Deployment

Deployment
import io
import os
import tarfile
import sagemaker

# 1. Save the PyTorch model to memory
buffer_model = io.BytesIO()
traced_model = torch.jit.script(model)
torch.jit.save(traced_model, buffer_model)

# 2. Upload to AWS S3
sagemaker_session = sagemaker.Session()
bucket_name_default = sagemaker_session.default_bucket()
model_name = f'tyre_quality_classifier.pth'

# 2.1. Zip PyTorch model into tar.gz file
buffer_zip = io.BytesIO()
with tarfile.open(fileobj=buffer_zip, mode="w:gz") as tar:
    # Add PyTorch pt file
    file_name = os.path.basename(model_name)
    file_name_with_extension = os.path.split(file_name)[-1]
tarinfo = tarfile.TarInfo(file_name_with_extension)
tarinfo.size = len(buffer_model.getbuffer())
buffer_model.seek(0)
tar.addfile(tarinfo, buffer_model)

# 2.2. Upload the tar.gz file to S3 bucket
buffer_zip.seek(0)
boto3.resource('s3') \ .Bucket(bucket_name_default) \ .Object(f'pytorch/{model_name}.tar.gz') \ .put(Body=buffer_zip.getvalue())

The code saves the PyTorch model to Amazon S3 because SageMaker requires the model to be stored in S3 for deployment. By uploading the model to Amazon S3, it becomes accessible to SageMaker, allowing for the deployment and inference on the deployed model.

import time
from sagemaker.pytorch import PyTorchModel
from sagemaker.predictor import Predictor
from sagemaker.serializers import IdentitySerializer
from sagemaker.deserializers import JSONDeserializer

class TyreQualitySerializer(IdentitySerializer):
def serialize(self, data):
    transformed_image = preprocess(data)
    tensor_image = torch.Tensor(transformed_image)
    serialized_data = io.BytesIO()
    torch.save(tensor_image, serialized_data)
    serialized_data.seek(0)
    serialized_data = serialized_data.read()
    return serialized_data

class TyreQualityPredictor(Predictor):
    def __init__(self, endpoint_name, sagemaker_session):
        super().__init__(endpoint_name, sagemaker_session=sagemaker_session,
                         serializer=TyreQualitySerializer(),
                         deserializer=JSONDeserializer(),
                         )

        sagemaker_model = PyTorchModel(
            model_data=f's3://{bucket_name_default}/pytorch/{model_name}.tar.gz',
            role=sagemaker.get_execution_role(),
            framework_version='2.0.1',
            py_version='py310',
            predictor_cls=TyreQualityPredictor,
            entry_point='inference.py',
            source_dir='code',
        )

        timestamp = int(time.time())
        pytorch_endpoint_name = '{:}-{:}-{:}'.format('tyre-quality-classifier', 'pt', timestamp)
        sagemaker_predictor = sagemaker_model.deploy(
            initial_instance_count=1,
            instance_type='ml.p3.2xlarge',
            endpoint_name=pytorch_endpoint_name
        )

This code facilitates the deployment of a PyTorch model on SageMaker. It defines a custom serializer, 
TyreQualitySerializer, which preprocesses and serializes input data as a PyTorch tensor. The 
TyreQualityPredictor class is a custom predictor that utilizes the defined serializer and a JSONDeserializer. 
The code also creates a PyTorchModel object to specify the model's S3 location, IAM role, framework version, 
and entry point for inference. The code generates a timestamp and constructs an endpoint name based on the
model and timestamp. Finally, the model is deployed using the deploy method, specifying the instance count, instance type, and generated endpoint name. This enables the PyTorch model to be deployed and accessible for inference on SageMaker.

**Inference**

```python
image_object = list(bucket.objects.filter('dataset/tyre'))[0].get()
image_bytes = image_object['Body'].read()

with Image.open(Image.open(BytesIO(image_bytes)) as image:
    predicted_classes = sagemaker_predictor.predict(image)

print(predicted_classes)
```

This is the example of using the deployed endpoint to do the inference.

**Part 3 - Building A Simplified MLOps Pipeline (CI/CT/CD)**

**Author(s):**
Jian Jian (Ken), Senior Data & Applied Scientist, NetApp

**Introduction**

In this tutorial, you will learn how to leverage various AWS services to construct a simple MLOps pipeline that encompasses Continuous Integration (CI), Continuous Training (CT), and Continuous Deployment (CD). Unlike traditional DevOps pipelines, MLOps requires additional considerations to complete the operational cycle. By following this tutorial, you will gain insights into incorporating CT into the MLOps loop, enabling continuous training of your models and seamless deployment for inference. The tutorial will guide you through the process of utilizing AWS services to establish this end-to-end MLOps pipeline.

**Manifest**

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data storage</td>
<td>AWS FSxN</td>
<td>Refer to Part 1 - Integrating AWS FSx for NetApp ONTAP (FSxN) as a private S3 bucket into AWS SageMaker.</td>
</tr>
<tr>
<td>Data science IDE</td>
<td>AWS SageMaker</td>
<td>This tutorial is based on the Jupyter notebook presented in Part 2 - Leveraging AWS FSx for NetApp ONTAP (FSxN) as a Data Source for Model Training in SageMaker.</td>
</tr>
<tr>
<td>Function to trigger the MLOps pipeline</td>
<td>AWS Lambda function</td>
<td>-</td>
</tr>
<tr>
<td>Cron job trigger</td>
<td>AWS EventBridge</td>
<td>-</td>
</tr>
<tr>
<td>Deep learning framework</td>
<td>PyTorch</td>
<td>-</td>
</tr>
<tr>
<td>AWS Python SDK</td>
<td>boto3</td>
<td>-</td>
</tr>
<tr>
<td>Functionality</td>
<td>Name</td>
<td>Comment</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td>---------</td>
</tr>
<tr>
<td>Programming Language</td>
<td>Python</td>
<td>v3.10</td>
</tr>
</tbody>
</table>

**Prerequisite**

- An pre-configured FSxN file system. This tutorial utilizes data stored in FSxN for the training process.
- A **SageMaker Notebook instance** that is configured to share the same VPC as the FSxN file system mentioned above.
- Before triggering the **AWS Lambda function**, ensure that the **SageMaker Notebook instance** is in **stopped** status.
- The **ml.g4dn.xlarge** instance type is required to leverage the GPU acceleration necessary for the computations of deep neural networks.

**Architecture**

This MLOps pipeline is a practical implementation that utilizes a cron job to trigger a serverless function, which in turn executes an AWS service registered with a lifecycle callback function. The **AWS EventBridge** acts as the cron job. It periodically invokes an **AWS Lambda function** responsible for retraining and redeploying the model. This process involves spinning up the **AWS SageMaker Notebook** instance to perform the necessary tasks.

**Step-by-Step Configuration**

**Lifecycle configurations**

To configure the lifecycle callback function for the AWS SageMaker Notebook instance, you would utilize **Lifecycle configurations**. This service allow you to define the necessary actions to be performed during when spinning up the notebook instance. Specifically, a shell script can be implemented within the **Lifecycle configurations** to automatically shut down the notebook instance once the training and deployment processes are completed. This is a required configuration as the cost is one of the major consideration in MLOps.

It’s important to note that the configuration for **Lifecycle configurations** needs to be set up in advance. Therefore, it is recommended to prioritize configuring this aspect before proceeding with the other MLOps pipeline setup.

1. To set up a Lifecycle configurations, open the **Sagemaker** panel and navigate to **Lifecycle configurations** under the section **Admin configurations**.
2. Select the **Notebook Instance** tab and click the **Create configuration** button.
3. Paste the below code to the entry area.

```bash
#!/bin/bash

set -e
sudo -u ec2-user -i <<'EOF'
# 1. Retraining and redeploying the model
NOTEBOOK_FILE=/home/ec2-user/SageMaker/tyre_quality_classification_local_training.ipynb
echo "Activating conda env"
source /home/ec2-user/anaconda3/bin/activate pytorch_p310
nohup jupyter nbconvert "$NOTEBOOK_FILE"
   --ExecutePreprocessor.kernel_name=python --execute --to notebook &
nbconvert_pid=$!
conda deactivate

# 2. Scheduling a job to shutdown the notebook to save the cost
PYTHON_DIR=/home/ec2-user/anaconda3/envs/JupyterSystemEnv/bin/python3.10'
echo "Starting the autostop script in cron"
   (crontab -l 2>/dev/null; echo "*/5 * * * * bash -c 'if ps -p $nbconvert_pid > /dev/null; then echo \"Notebook is still running.\" >> /var/log/jupyter.log; else echo \"Notebook execution completed.\" >> /var/log/jupyter.log; $PYTHON_DIR -c \"import boto3; boto3.client( \"sagemaker\")\".stop_notebook_instance(NotebookInstanceName=get_notebook_name())\" >> /var/log/jupyter.log; fi')") | crontab -
EOF
```
4. This script executes the Jupyter Notebook, which handles the retraining and redeployment of the model for inference. After the execution is complete, the notebook will automatically shut down within 5 minutes. To learn more about the problem statement and the code implementation, please refer to Part 2 - Leveraging AWS FSx for NetApp ONTAP (FSxN) as a Data Source for Model Training in SageMaker.

5. After the creation, navigate to Notebook instances, select the target instance, and click **Update settings** under Actions dropdown.
6. Select the created Lifecycle configuration and click Update notebook instance.
AWS Lambda serverless function

As mentioned earlier, the **AWS Lambda function** is responsible for spinning up the **AWS SageMaker Notebook instance**.

1. To create an **AWS Lambda function**, navigate to the respective panel, switch to the **Functions** tab, and click on **Create Function**.

2. Please file all required entries on the page and remember to switch the **Runtime** to **Python 3.10**.
3. Please verify that the designated role has the required permission **AmazonSageMakerFullAccess** and click on the **Create function** button.
4. Select the created Lambda function. In the code tab, copy and paste the following code into the text area. This code starts the notebook instance named `fsxn-ontap`.

```python
import boto3
import logging

def lambda_handler(event, context):
    client = boto3.client('sagemaker')
    logging.info('Invoking SageMaker')
    client.start_notebook_instance(NotebookInstanceName='fsxn-ontap')
    return {
        'statusCode': 200,
        'body': f'Starting notebook instance: {notebook_instance_name}'
    }
```
5. Click the **Deploy** button to apply this code change.

6. To specify how to trigger this AWS Lambda function, click on the Add Trigger button.
7. Select EventBridge from the dropdown menu, then click on the radio button labeled Create a new rule. In the schedule expression field, enter \texttt{rate(1 day)}\texttt{,} and click on the Add button to create and apply this new cron job rule to the AWS Lambda function.
After completing the two-step configuration, on a daily basis, the **AWS Lambda function** will initiate the **SageMaker Notebook**, perform model retraining using the data from the **FSxN** repository, redeploy the updated model to the production environment, and automatically shut down the **SageMaker Notebook instance** to optimize cost. This ensures that the model remains up to date.

This concludes the tutorial for developing an MLOps pipeline.

**Hybrid Multicloud MLOps with Domino Data Lab and NetApp**
Hybrid Multicloud MLOps with Domino Data Lab and NetApp

Mike Oglesby, NetApp

Organizations all over the world are currently adopting AI to transform their businesses and processes. Because of this, AI-ready compute infrastructure is often in short supply. Enterprises are adopting hybrid multicloud MLOps architectures in order to take advantage of available compute environments across different regions, data centers, and clouds - balancing cost, availability, and performance.

Domino Nexus, from Domino Data Lab, is a unified MLOps control plane that lets you run data science and machine learning workloads across any compute cluster — in any cloud, region, or on-premises. It unifies data science silos across the enterprise, so you have one place to build, deploy, and monitor models. Likewise, NetApp’s hybrid cloud data management capabilities enable you to bring your data to your jobs and workspaces, no matter where they are running. When you pair Domino Nexus with NetApp, you have the flexibility to schedule workloads across environments without having to worry about data availability. In other words, you have the ability to send your workloads and your data to the appropriate compute environment, enabling you to accelerate your AI deployments while navigating regulations around data privacy and sovereignty.

This solution demonstrates the deployment of a unified MLOps control plane incorporating an on-premises Kubernetes cluster and an Elastic Kubernetes Service (EKS) cluster running in Amazon Web Services (AWS).

Technology Overview

Domino Data Lab

Domino Data Lab powers model-driven businesses with its leading Enterprise AI platform trusted by over 20% of the Fortune 100. Domino accelerates the development and deployment of data science work while increasing collaboration and governance. With Domino, enterprises worldwide can develop better medicines, grow more productive crops, build better cars, and much more. Founded in 2013, Domino is backed by Coatue Management, Great Hill Partners, Highland Capital, Sequoia Capital and other leading investors.

Domino lets enterprises and their data scientists build, deploy and manage AI on a unified, end-to-end platform
— fast, responsibly and cost-effectively. Teams can access all of the data, tools, compute, models, and projects they need across any environment, so they can collaborate, reuse past work, track models in production to improve accuracy, standardize with best practices, and make AI responsible and governed.

- **Open and Flexible**: Access the broadest ecosystem of open source and commercial tools, and infrastructure, for the best innovations and no vendor lock-in.
- **System of Record**: Central hub for AI operations and knowledge across the enterprise, enabling best practices, cross-functional collaboration, faster innovation, and efficiency.
- **Integrated**: Integrated workflows and automation — built for enterprise processes, controls, and governance — satisfy your compliance and regulatory needs.
- **Hybrid Multicloud**: Run AI workloads close to your data anywhere — on-premises, hybrid, any cloud or multi-cloud — for lower cost, optimal performance and compliance.

**Domino Nexus**

Domino Nexus is a single pane of glass that lets you run data science and machine learning workloads across any compute cluster — in any cloud, region, or on-premises. It unifies data science silos across the enterprise, so you have one place to build, deploy, and monitor models.

**NetApp BlueXP**

NetApp BlueXP unifies all of NetApp’s storage and data services into a single tool that lets you build, protect, and govern your hybrid multicloud data estate. It delivers a unified experience for storage and data services across on-premises and cloud environments, and enables operational simplicity through the power of AIOps, with the flexible consumption parameters and integrated protection required for today’s cloud-led world.
NetApp ONTAP

ONTAP 9, the latest generation of storage management software from NetApp, enables businesses to modernize infrastructure and transition to a cloud-ready data center. Leveraging industry-leading data management capabilities, ONTAP enables the management and protection of data with a single set of tools, regardless of where that data resides. You can also move data freely to wherever it is needed: the edge, the core, or the cloud. ONTAP 9 includes numerous features that simplify data management, accelerate, and protect critical data, and enable next generation infrastructure capabilities across hybrid cloud architectures.

Simplify data management

Data management is crucial to enterprise IT operations and data scientists so that appropriate resources are used for AI applications and training AI/ML datasets. The following additional information about NetApp technologies is out of scope for this validation but might be relevant depending on your deployment.

ONTAP data management software includes the following features to streamline and simplify operations and reduce your total cost of operation:

• Inline data compaction and expanded deduplication. Data compaction reduces wasted space inside storage blocks, and deduplication significantly increases effective capacity. This applies to data stored locally and data tiered to the cloud.

• Minimum, maximum, and adaptive quality of service (AQoS). Granular quality of service (QoS) controls help maintain performance levels for critical applications in highly shared environments.

• NetApp FabricPool. Provides automatic tiering of cold data to public and private cloud storage options, including Amazon Web Services (AWS), Azure, and NetApp StorageGRID storage solution. For more information about FabricPool, see TR-4598: FabricPool best practices.

Accelerate and protect data

ONTAP delivers superior levels of performance and data protection and extends these capabilities in the following ways:

• Performance and lower latency. ONTAP offers the highest possible throughput at the lowest possible latency.

• Data protection. ONTAP provides built-in data protection capabilities with common management across all platforms.

• NetApp Volume Encryption (NVE). ONTAP offers native volume-level encryption with both onboard and External Key Management support.

• Multitenancy and multifactor authentication. ONTAP enables sharing of infrastructure resources with the highest levels of security.

Future-proof infrastructure

ONTAP helps meet demanding and constantly changing business needs with the following features:

• Seamless scaling and nondisruptive operations. ONTAP supports the nondisruptive addition of capacity to existing controllers and to scale-out clusters. Customers can upgrade to the latest technologies, such as NVMe and 32Gb FC, without costly data migrations or outages.

• Cloud connection. ONTAP is the most cloud-connected storage management software, with options for software-defined storage and cloud-native instances in all public clouds.

• Integration with emerging applications. ONTAP offers enterprise-grade data services for next generation platforms and applications, such as autonomous vehicles, smart cities, and Industry 4.0, by using the same
infrastructure that supports existing enterprise apps.

Amazon FSx for NetApp ONTAP

Amazon FSx for NetApp ONTAP is a first-party, fully managed AWS service that provides highly reliable, scalable, high-performing, and feature-rich file storage built on NetApp's popular ONTAP file system. FSx for ONTAP combines the familiar features, performance, capabilities, and API operations of NetApp file systems with the agility, scalability, and simplicity of a fully managed AWS service.

NetApp Astra Trident

Astra Trident enables consumption and management of storage resources across all popular NetApp storage platforms, in the public cloud or on premises, including ONTAP (AFF, FAS, Select, Cloud, Amazon FSx for NetApp ONTAP), Element software (NetApp HCI, SolidFire), Azure NetApp Files service, and Cloud Volumes Service on Google Cloud. Astra Trident is a Container Storage Interface (CSI) compliant dynamic storage orchestrator that natively integrates with Kubernetes.

Kubernetes

Kubernetes is an open source, distributed, container orchestration platform that was originally designed by Google and is now maintained by the Cloud Native Computing Foundation (CNCF). Kubernetes enables the automation of deployment, management, and scaling functions for containerized applications, and is the dominant container orchestration platform in enterprise environments.

Amazon Elastic Kubernetes Service (EKS)

Amazon Elastic Kubernetes Service (Amazon EKS) is a managed Kubernetes service in the AWS cloud. Amazon EKS automatically manages the availability and scalability of the Kubernetes control plane nodes responsible for scheduling containers, managing application availability, storing cluster data, and other key tasks. With Amazon EKS, you can take advantage of all the performance, scale, reliability, and availability of AWS infrastructure, as well as integrations with AWS networking and security services.

Architecture

This solution combines Domino Nexus' hybrid multicloud workload scheduling capabilities with NetApp data services to create a unified hybrid cloud MLOps platform. See the following table for details.

<table>
<thead>
<tr>
<th>Component</th>
<th>Name</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLOps Control Plane</td>
<td>Domino Enterprise AI Platform with Domino Nexus</td>
<td>AWS</td>
</tr>
<tr>
<td>MLOps Platform Compute Environments</td>
<td>Domino Nexus Data Planes</td>
<td>AWS, On-premises data center</td>
</tr>
<tr>
<td>On-premises Compute Platform</td>
<td>Kubernetes with NetApp Astra Trident</td>
<td>On-premises data center</td>
</tr>
<tr>
<td>Cloud Compute Platform</td>
<td>Amazon Elastic Kubernetes Service (EKS) with NetApp Astra Trident</td>
<td>AWS</td>
</tr>
<tr>
<td>On-premises Data Platform</td>
<td>NetApp storage appliance powered by NetApp ONTAP</td>
<td>On-premises data center</td>
</tr>
<tr>
<td>Cloud Data Platform</td>
<td>Amazon FSx for NetApp ONTAP</td>
<td>AWS</td>
</tr>
</tbody>
</table>
Initial Setup

This section describes the initial setup tasks that need to be performed in order to utilize Domino Nexus with NetApp data services in a hybrid environment incorporating an on-premises data center and AWS.

Prerequisites

Before you perform the steps that are outlined in this section, we assume that you have already performed the following tasks:

- You have already deployed and configured your on-premises NetApp ONTAP storage platform. For more information, refer to the NetApp product documentation.
- You have already provisioned an Amazon FSx for NetApp ONTAP instance in AWS. For more information, refer to the Amazon FSx for NetApp ONTAP product page.
- You have already provisioned a Kubernetes cluster in your on-premises data center. For more information, refer to the Domino admin guide.
- You have installed NetApp Astra Trident in your on-premises Kubernetes cluster. Additionally, you have configured this Trident instance to use your on-premises NetApp ONTAP storage platform when provisioning and managing storage resources. For more information, refer to the NetApp Astra Trident documentation.
- You have installed NetApp Astra Trident in your Amazon EKS cluster. Additionally, you have configured this Trident instance to use your Amazon FSx for NetApp ONTAP instance when provisioning and managing storage resources. For more information, refer to the NetApp Astra Trident documentation.
- You must have bi-directional network connectivity between your on-premises data center and your Virtual Private Cloud (VPC) in AWS. For more details on the various options for implementing this, refer to the Amazon Virtual Private Network (VPN) documentation.
Install the Domino Enterprise AI Platform in AWS

To install the Domino Enterprise MLOps Platform in AWS, follow the instructions outlined in Domino admin guide. You must deploy Domino in the same Amazon EKS cluster that you previously provisioned. Additionally, NetApp Astra Trident must already be installed and configured in this EKS cluster, and you must specify a Trident-managed storage class as the shared storage class in your domino.yml install configuration file.

Refer to the Domino install configuration reference guide for details on how to specify a shared storage class in your domino.yml install configuration file.

Technical Report TR-4952 walks through the deployment of Domino in AWS with Amazon FSx for NetApp ONTAP and may be a useful reference for troubleshooting any issues that arise.

Enable Domino Nexus

Next, you must enable Domino Nexus. Refer to the Domino admin guide for details.

Deploy a Domino Data Plane in your On-premises Data Center

Next, you must deploy a Domino Data Plane in your on-premises data center. You must deploy this data plane in the on-premises Kubernetes cluster that you previously provisioned. Additionally, NetApp Astra Trident must already be installed and configured in this Kubernetes cluster. Refer to the Domino admin guide for details.

Expose Existing NetApp Volumes to Domino

This section describes the tasks that need to be performed in order to expose existing NetApp ONTAP NFS volumes to the Domino MLOps platform. These same steps apply both on-premises and in AWS.

Why Expose NetApp ONTAP Volumes to Domino?

Using NetApp volumes in conjunction with Domino provides the following benefits:

• You can execute workloads against extremely large datasets by taking advantage of NetApp ONTAP’s scale-out capabilities.
• You can execute workloads across multiple compute nodes without having to copy your data to the individual nodes.
• You can take advantage of NetApp’s hybrid multicloud data movement and sync capabilities in order to access your data across multiple data centers and/or clouds.
• You want to be able to quickly and easily create a cache of your data in a different data center or cloud.

Expose Existing NFS Volumes that were not Provisioned by Astra Trident

If your existing NetApp ONTAP NFS volume was not provisioned by Astra Trident, follow the steps outlined in this sub-section.

Create PV and PVC in Kubernetes

For on-premises volumes, create the PV and PVC in your on-premises Kubernetes cluster. For Amazon FSx for NetApp ONTAP volumes, create the PV and PVC in Amazon EKS.
First, you must create a persistent volume (PV) and persistent volume claim (PVC) in your Kubernetes cluster. To create the PV and PVC, use the NFS PV/PVC example from the Domino admin guide and update the values to reflect to your environment. Be sure to specify the correct values for the namespace, nfs.path, and nfs.server fields. Additionally, we recommend giving your PV and PVC unique names that represent that nature of the data that is stored on the corresponding ONTAP NFS volume. For example, if the volume contains images of manufacturing defects, you might name the PV, `pv-mfg-defect-images`, and the PVC, `pvc-mfg-defect-images`.

**Register External Data Volume in Domino**

Next, you must register an external data volume in Domino. To register an external data volume, refer to the instructions in the Domino admin guide. When registering the volume, be sure to select "NFS" from the 'Volume Type' drop-down menu. After selecting "NFS", you should see your PVC in the 'Available Volumes' list.

![Register an External Volume](image)

**Expose Existing Volumes that were Provisioned by Astra Trident**

If your existing volume was provisioned by Astra Trident, follow the steps outlined in this sub-section.
Edit Existing PVC

If your volume was provisioned by Astra Trident, then you already have a persistent volume claim (PVC) corresponding to your volume. In order to expose this volume to Domino, you must edit the PVC and add the following label to the list of labels in the `metadata.labels` field:

```
"dominodatalab.com/external-data-volume": "Generic"
```

Register External Data Volume in Domino

Next, you must register an external data volume in Domino. To register an external data volume, refer to the instructions in the Domino admin guide. When registering the volume, be sure to select "Generic" from the 'Volume Type' drop-down menu. After selecting "Generic", you should see your PVC in the 'Available Volumes' list.

Access the same Data Across Different Environments

This section describes the tasks that need to be performed in order to access the same data across different compute environments. In the Domino MLOps platform, compute environments are referred to "data planes." Follow the tasks outlined in this section if your data resides on a NetApp volume in one data plane, but you need to access it in another data plane. This type of scenario is often referred to as "bursting" or, when the destination environment is the cloud, "cloud bursting." This capability is often needed when dealing with constrained or over-subscribed compute resources. For example, if your on-premises compute cluster is over-subscribed, you may want to schedule workloads to the cloud where they can be started immediately.

There are two recommended options for accessing a NetApp volume that resides in a different data plane. These options are outlined in the sub-sections below. Choose one of these options depending on your specific requirements. The benefits and drawbacks of the two options are described in the following table.

<table>
<thead>
<tr>
<th>Option</th>
<th>Benefits</th>
<th>Drawbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option 1 - Cache</td>
<td>- Simpler workflow&lt;br&gt;- Ability to cache a subset of data based on needs&lt;br&gt;- Ability to write data back to source&lt;br&gt;- No remote copy to manage</td>
<td>- Increased latency on initial data access as cache is hydrated.</td>
</tr>
<tr>
<td>Option 2 - Mirror</td>
<td>- Full copy of source volume&lt;br&gt;- No increased latency due to cache hydration (after mirror operation is complete)</td>
<td>- Must wait for mirror operation to complete before accessing data&lt;br&gt;- Must manage a remote copy&lt;br&gt;- No ability to write back to source</td>
</tr>
</tbody>
</table>

Option 1 - Create a Cache of a Volume that Resides in a Different Data Plane

With NetApp FlexCache technology, you can create a cache of a NetApp volume that resides in a different data plane. For example, if you have a NetApp volume in your on-premises data plane, and you need to access that volume in your AWS data plane, you can create a cache of the volume in AWS. This section outlines the tasks that need to be performed in order to create a cache of a NetApp volume that resides in a different data plane.
Create FlexCache Volume in Destination Environment

If the destination environment is your on-premises data center, you will create the FlexCache volume on your on-premises ONTAP system. If the destination environment is AWS, you will create the FlexCache volume on your Amazon FSx for NetApp ONTAP instance.

First, you must create a FlexCache volume in the destination environment.

We recommend using BlueXP to create the FlexCache volume. To create a FlexCache volume with BlueXP, follow the instructions outlined in the BlueXP volume caching documentation.

If you prefer not to use BlueXP, you can use ONTAP System Manager or the ONTAP CLI to create the FlexCache volume. To create a FlexCache volume with System Manager, refer to the instructions outlined in the ONTAP documentation. To create a FlexCache volume with the ONTAP CLI, refer to the instructions outlined in the ONTAP documentation.

If you wish to automate this process, you can use the BlueXP API, the ONTAP REST API, or the ONTAP Ansible collection.

 System Manager is not available in Amazon FSx for NetApp ONTAP.

Expose FlexCache Volume to Domino

Next, you must expose the FlexCache volume to the Domino MLOps platform. To expose the FlexCache volume to Domino, follow the instructions outlined in the 'Expose Existing NFS Volumes that were not Provisioned by Astra Trident' sub-section of the 'Expose Existing NetApp Volumes to Domino' section of this solution.

Now, you will be able to mount the FlexCache volume when launching jobs and workspaces in the destination data plane as shown in the following screenshots.
After Exposing FlexCache Volume to Domino
Option 2 - Replicate a Volume that Resides in a Different Data Plane

With NetApp SnapMirror data replication technology, you can create a copy of a NetApp volume that resides in a different data plane. For example, if you have a NetApp volume in your on-premises data plane, and you need to access that volume in your AWS data plane, you can create a copy of the volume in AWS. This section outlines the tasks that need to be performed in order to create a copy of a NetApp volume that resides in a different data plane.

Create SnapMirror Relationship

First, you must create a SnapMirror relationship between your source volume and a new destination volume in the destination environment. Note that the destination volume will be created as part of the process of creating the SnapMirror relationship.
We recommend using BlueXP to create the SnapMirror relationship. To create a SnapMirror relationship with BlueXP, follow the instructions outlined in the BlueXP replication documentation.

If you prefer not to use BlueXP, you can use ONTAP System Manager or the ONTAP CLI to create the SnapMirror relationship. To create a SnapMirror relationship with System Manager, refer to the instructions outlined in the ONTAP documentation. To create a SnapMirror relationship with the ONTAP CLI, refer to the instructions outlined in the ONTAP documentation.

If you wish to automate this process, you can use the BlueXP API, the ONTAP REST API, or the ONTAP Ansible collection.

System Manager is not available in Amazon FSx for NetApp ONTAP.

Break SnapMirror Relationship

Next, you must break the SnapMirror relationship in order to activate the destination volume for data access. Wait until the initial replication is complete before performing this step.

You can determine whether or not the replication is complete by checking the mirror state in BlueXP, ONTAP System Manager, or the ONTAP CLI. When the replication is complete, the mirror state will be "snapmirrored".

We recommend using BlueXP to break the SnapMirror relationship. To break a SnapMirror relationship with BlueXP, follow the instructions outlined in the BlueXP replication documentation.

If you prefer not to use BlueXP, you can use ONTAP System Manager or the ONTAP CLI to break the SnapMirror relationship. To break a SnapMirror relationship with System Manager, refer to the instructions outlined in the ONTAP documentation. To break a SnapMirror relationship with the ONTAP CLI, refer to the instructions outlined in the ONTAP documentation.

If you wish to automate this process, you can use the BlueXP API, the ONTAP REST API, or the ONTAP Ansible collection.

Expose Destination Volume to Domino

Next, you must expose the destination volume to the Domino MLOps platform. To expose the destination volume to Domino, follow the instructions outlined in the 'Expose Existing NFS Volumes that were not Provisioned by Astra Trident' sub-section of the 'Expose Existing NetApp Volumes to Domino' section of this solution.

Now, you will be able to mount the destination volume when launching jobs and workspaces in the destination data plane as shown in the following screenshots.

Before Creating SnapMirror Relationship
After Exposing Destination Volume to Domino
Where to Find Additional Information

To learn more about the information described in this document, refer to the following documents and/or websites:

- Domino Data Lab
  https://domino.ai
- Domino Nexus
  https://domino.ai/platform/nexus
Acknowledgments

- Josh Mineroff, Director of SA for Tech Alliances, Domino Data Lab
- Nicholas Jablonski, Field CTO, Domino Data Lab
- Prabu Arjunan, Solution Architect, NetApp
- Brian Young, Global Alliance Director, Technology Alliance Partners, NetApp

NVIDIA AI Enterprise with NetApp and VMware

NVIDIA AI Enterprise with NetApp and VMware

Mike Oglesby, NetApp

For IT architects and admins, AI tooling can be complicated and unfamiliar. Additionally, many AI platforms are not enterprise-ready. NVIDIA AI Enterprise, powered by NetApp and VMware, was created to deliver a streamlined, enterprise-class AI architecture.

NVIDIA AI Enterprise is an end-to-end, cloud-native suite of AI and data analytics software that is optimized, certified, and supported by NVIDIA to run on VMware vSphere with NVIDIA-Certified Systems. This software facilitates the simple and rapid deployment, management, and scaling of AI workloads in the modern hybrid cloud environment. NVIDIA AI Enterprise, powered by NetApp and VMware, delivers enterprise-class AI workload and data management in a simplified, familiar package.
Technology Overview

NVIDIA AI Enterprise

NVIDIA AI Enterprise is an end-to-end, cloud-native suite of AI and data analytics software that is optimized, certified, and supported by NVIDIA to run on VMware vSphere with NVIDIA-Certified Systems. This software facilitates the simple and rapid deployment, management, and scaling of AI workloads in the modern hybrid cloud environment.

NVIDIA GPU Cloud (NGC)

NVIDIA NGC hosts a catalog of GPU-optimized software for AI practitioners to develop their AI solutions. It also provides access to various AI services including NVIDIA Base Command for model training, NVIDIA Fleet Command to deploy and monitor models, and the NGC Private Registry for securely accessing and managing proprietary AI software. Also, NVIDIA AI Enterprise customers can request support through the NGC portal.

VMware vSphere

VMware vSphere is VMware’s virtualization platform, which transforms data centers into aggregated computing infrastructures that include CPU, storage, and networking resources. vSphere manages these infrastructures as a unified operating environment, and provides administrators with the tools to manage the data centers that participate in that environment.

The two core components of vSphere are ESXi and vCenter Server. ESXi is the virtualization platform where administrators create and run virtual machines and virtual appliances. vCenter Server is the service through which administrators manage multiple hosts connected in a network and pool host resources.
ONTAP 9, the latest generation of storage management software from NetApp, enables businesses to modernize infrastructure and transition to a cloud-ready data center. Leveraging industry-leading data management capabilities, ONTAP enables the management and protection of data with a single set of tools, regardless of where that data resides. You can also move data freely to wherever it is needed: the edge, the core, or the cloud. ONTAP 9 includes numerous features that simplify data management, accelerate, and protect critical data, and enable next generation infrastructure capabilities across hybrid cloud architectures.

**Simplify data management**

Data management is crucial to enterprise IT operations and data scientists so that appropriate resources are used for AI applications and training AI/ML datasets. The following additional information about NetApp technologies is out of scope for this validation but might be relevant depending on your deployment.

ONTAP data management software includes the following features to streamline and simplify operations and reduce your total cost of operation:

- Inline data compaction and expanded deduplication. Data compaction reduces wasted space inside storage blocks, and deduplication significantly increases effective capacity. This applies to data stored locally and data tiered to the cloud.

- Minimum, maximum, and adaptive quality of service (AQoS). Granular quality of service (QoS) controls help maintain performance levels for critical applications in highly shared environments.

- NetApp FabricPool. Provides automatic tiering of cold data to public and private cloud storage options, including Amazon Web Services (AWS), Azure, and NetApp StorageGRID storage solution. For more information about FabricPool, see TR-4598: FabricPool best practices.

**Accelerate and protect data**

ONTAP delivers superior levels of performance and data protection and extends these capabilities in the following ways:

- Performance and lower latency. ONTAP offers the highest possible throughput at the lowest possible latency.

- Data protection. ONTAP provides built-in data protection capabilities with common management across all platforms.

- NetApp Volume Encryption (NVE). ONTAP offers native volume-level encryption with both onboard and External Key Management support.

- Multitenancy and multifactor authentication. ONTAP enables sharing of infrastructure resources with the highest levels of security.

**Future-proof infrastructure**

ONTAP helps meet demanding and constantly changing business needs with the following features:

- Seamless scaling and nondisruptive operations. ONTAP supports the nondisruptive addition of capacity to existing controllers and to scale-out clusters. Customers can upgrade to the latest technologies, such as NVMe and 32Gb FC, without costly data migrations or outages.

- Cloud connection. ONTAP is the most cloud-connected storage management software, with options for software-defined storage (ONTAP Select) and cloud-native instances (NetApp Cloud Volumes Service) in all public clouds.
• Integration with emerging applications. ONTAP offers enterprise-grade data services for next generation platforms and applications, such as autonomous vehicles, smart cities, and Industry 4.0, by using the same infrastructure that supports existing enterprise apps.

NetApp DataOps Toolkit

The NetApp DataOps Toolkit is a Python-based tool that simplifies the management of development/training workspaces and inference servers that are backed by high-performance, scale-out NetApp storage. Key capabilities include:

• Rapidly provision new high-capacity JupyterLab workspaces that are backed by high-performance, scale-out NetApp storage.
• Rapidly provision new NVIDIA Triton Inference Server instances that are backed by enterprise-class NetApp storage.
• Near-instantaneously clone high-capacity JupyterLab workspaces in order to enable experimentation or rapid iteration.
• Near-instantaneously save snapshots of high-capacity JupyterLab workspaces for backup and/or traceability/baselining.
• Near-instantaneously provision, clone, and snapshot high-capacity, high-performance data volumes.

Architecture

This solution builds upon a proven and familiar architecture featuring NetApp, VMware, and NVIDIA-Certified Systems. See the following table for details.

<table>
<thead>
<tr>
<th>Component</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI and Data Analytics Software</td>
<td>NVIDIA AI Enterprise for VMware</td>
</tr>
<tr>
<td>Virtualization Platform</td>
<td>VMware vSphere</td>
</tr>
<tr>
<td>Compute Platform</td>
<td>NVIDIA-Certified Systems</td>
</tr>
<tr>
<td>Data Management Platform</td>
<td>NetApp ONTAP</td>
</tr>
</tbody>
</table>
Initial Setup

This section describes the initial setup tasks that need to be performed in order to utilize NVIDIA AI Enterprise with NetApp and VMware.

Prerequisites

Before you perform the steps that are outlined in this section, we assume that you have already deployed VMware vSphere and NetApp ONTAP. Refer to the NVIDIA AI Enterprise Product Support Matrix for details on supported vSphere versions. Refer to the NetApp and VMware solution documentation for details on deploying VMware vSphere with NetApp ONTAP.
Install NVIDIA AI Enterprise Host Software

To install the NVIDIA AI Enterprise host software, follow the instructions outlined in sections 1-4 in the NVIDIA AI Enterprise Quick Start Guide.

Utilize NVIDIA NGC Software

This section describes the tasks that need to be performed in order to utilize NVIDIA NGC enterprise software within an NVIDIA AI Enterprise environment.

Setup

This section describes the initial setup tasks that need to be performed in order to utilize NVIDIA NGC enterprise software within an NVIDIA AI Enterprise environment.

Prerequisites

Before you perform the steps that are outlined in this section, we assume that you have already deployed the NVIDIA AI Enterprise host software by following the instructions outlined on the Initial Setup page.

Create an Ubuntu Guest VM with vGPU

First, you must create an Ubuntu 20.04 guest VM with vGPU. To create an Ubuntu 20.04 guest VM with vGPU, follow the instructions outline in the NVIDIA AI Enterprise Deployment Guide.

Download and Install NVIDIA Guest Software

Next, you must install the required NVIDIA guest software within the guest VM that you created in the previous step. To download and install the required NVIDIA guest software within the guest VM, follow the instructions outlined in sections 5.1-5.4 in the NVIDIA AI Enterprise Quick Start Guide.

When performing the verification tasks outlined in section 5.4, you may need to use a different CUDA container image version tag as the CUDA container image has been updated since the writing of the guide. In our validation, we used 'nvidia/cuda:11.0.3-base-ubuntu20.04'.

Download AI/Analytics Framework Container(s)

Next, you must download needed AI or analytics framework container images from NVIDIA NGC so that they will be available within your guest VM. To download framework containers within the guest VM, follow the instructions outlined in the NVIDIA AI Enterprise Deployment Guide.

Install and Configure the NetApp DataOps Toolkit

Next, you must install the NetApp DataOps Toolkit for Traditional Environments within the guest VM. The NetApp DataOps Toolkit can be used to manage scale-out data volumes on your ONTAP system directly from the terminal within the guest VM. To install the NetApp DataOps Toolkit within the guest VM, perform the following tasks.

1. Install pip.
$ sudo apt update
$ sudo apt install python3-pip
$ python3 -m pip install netapp-dataops-traditional

2. Log out of the guest VM terminal and then log back in.

3. Configure the NetApp DataOps Toolkit. In order to complete this step, you will need API access details for your ONTAP system. You may need to obtain these from your storage admin.

$ netapp_dataops_cli.py config

Enter ONTAP management LIF hostname or IP address (Recommendation: Use SVM management interface): 172.22.10.10
Enter SVM (Storage VM) name: NVAIE-client
Enter SVM NFS data LIF hostname or IP address: 172.22.13.151
Enter default volume type to use when creating new volumes (flexgroup/flexvol) [flexgroup]:
Enter export policy to use by default when creating new volumes [default]:
Enter snapshot policy to use by default when creating new volumes [none]:
Enter unix filesystem user id (uid) to apply by default when creating new volumes (ex. '0' for root user) [0]:
Enter unix filesystem group id (gid) to apply by default when creating new volumes (ex. '0' for root group) [0]:
Enter unix filesystem permissions to apply by default when creating new volumes (ex. '0777' for full read/write permissions for all users and groups) [0777]:
Enter aggregate to use by default when creating new FlexVol volumes: aff_a400_01_NVME_SSD_1
Enter ONTAP API username (Recommendation: Use SVM account): admin
Enter ONTAP API password (Recommendation: Use SVM account): 
Verify SSL certificate when calling ONTAP API (true/false): false
Do you intend to use this toolkit to trigger BlueXP Copy and Sync operations? (yes/no): no
Do you intend to use this toolkit to push/pull from S3? (yes/no): no
Created config file: '/home/user/.netapp_dataops/config.json'.

Create a Guest VM template

Lastly, you must create a VM template based on your guest VM. You will be able to use this template to quickly create guest VMs for utilizing NVIDIA NGC software.

To create a VM template based on your guest VM, log into VMware vSphere, right-click on the guest VM name, choose 'Clone', choose 'Clone to Template...', and then follow the wizard.
Example Use Case - TensorFlow Training Job

This section describes the tasks that need to be performed in order to execute a TensorFlow training job within an NVIDIA AI Enterprise environment.

Prerequisites

Before you perform the steps that are outlined in this section, we assume that you have already created a guest VM template by following the instructions outlined on the Setup page.

Create Guest VM from Template

First, you must create a new guest VM from the template that you created in the previous section. To create a new guest VM from your template, log into VMware vSphere, right-click on the template name, choose 'New VM from This Template…', and then follow the wizard.
Create and Mount Data Volume

Next, you must create a new data volume on which to store your training dataset. You can quickly create a new data volume using the NetApp DataOps Toolkit. The example command that follows shows the creation of a volume named 'imagenet' with a capacity of 2 TB.

$ netapp_dataops_cli.py create vol -n imagenet -s 2TB

Before you can populate your data volume with data, you must mount it within the guest VM. You can quickly mount a data volume using the NetApp DataOps Toolkit. The example command that follows shows the mounting of the volume that was created in the previous step.

$ sudo -E netapp_dataops_cli.py mount vol -n imagenet -m ~/imagenet

Populate Data Volume

After the new volume has been provisioned and mounted, the training dataset can be retrieved from the source location and placed on the new volume. This typically will involve pulling the data from an S3 or Hadoop data lake and sometimes will involve help from a data engineer.

Execute TensorFlow Training Job

Now, you are ready to execute your TensorFlow training job. To execute your TensorFlow training job, perform the following tasks.

1. Pull the NVIDIA NGC enterprise TensorFlow container image.

   $ sudo docker pull nvcr.io/nvaie/tensorflow-2-1:22.05-tf1-nvaie-2.1-py3

2. Launch an instance of the NVIDIA NGC enterprise TensorFlow container. Use the '-v' option to attach your data volume to the container.

   $ sudo docker run --gpus all -v ~/imagenet:/imagenet -it --rm nvcr.io/nvaie/tensorflow-2-1:22.05-tf1-nvaie-2.1-py3

3. Execute your TensorFlow training program within the container. The example command that follows shows the execution of an example ResNet-50 training program that is included in the container image.

   $ python ./nvidia-examples/cnn/resnet.py --layers 50 -b 64 -i 200 -u batch --precision fp16 --data_dir /imagenet/data

Where to Find Additional Information

To learn more about the information described in this document, refer to the following documents and/or websites:
TR-4851: NetApp StorageGRID data lake for autonomous driving workloads - Solution design

David Arnette, NetApp

TR-4851 demonstrates the use of NetApp StorageGRID object storage as a data repository and management system for machine learning (ML) and deep learning (DL) software development. This paper describes the data flow and requirements in autonomous vehicle software development and the StorageGRID features that streamline the data lifecycle. This solution applies to any multistage data pipeline workflow that is typical in ML and DL development processes.

NetApp AI Control Plane

TR-4798: NetApp AI Control Plane

Mike Oglesby, NetApp

Companies and organizations of all sizes and across many industries are turning to artificial intelligence (AI), machine learning (ML), and deep learning (DL) to solve real-world problems, deliver innovative products and services, and to get an edge in an increasingly competitive marketplace. As organizations increase their use of AI, ML, and DL, they face many challenges, including workload scalability and data availability. This document demonstrates how you can address these challenges by using the NetApp AI Control Plane, a solution that pairs NetApp data management capabilities with popular open-source tools and frameworks.

This report shows you how to rapidly clone a data namespace. It also shows you how to seamlessly replicate data across sites and regions to create a cohesive and unified AI/ML/DL data pipeline. Additionally, it walks you through the defining and implementing of AI, ML, and DL training workflows that incorporate the near-instant
creation of data and model baselines for traceability and versioning. With this solution, you can trace every model training run back to the exact dataset that was used to train and/or validate the model. Lastly, this document shows you how to swiftly provision Jupyter Notebook workspaces with access to massive datasets.

Note: For HPC style distributed training at scale involving a large number of GPU servers that require shared access to the same dataset, or if you require/prefer a parallel file system, check out TR-4890. This technical report describes how to include NetApp's fully supported parallel file system solution BeeGFS as part of the NetApp AI Control Plane. This solution is designed to scale from a handful of NVIDIA DGX A100 systems, up to a full blown 140 node SuperPOD.

The NetApp AI Control Plane is targeted towards data scientists and data engineers, and, thus, minimal NetApp or NetApp ONTAP® expertise is required. With this solution, data management functions can be executed using simple and familiar tools and interfaces. If you already have NetApp storage in your environment, you can test drive the NetApp AI Control plane today. If you want to test drive the solution but you do not have already have NetApp storage, visit cloud.netapp.com, and you can be up and running with a cloud-based NetApp storage solution in minutes. The following figure provides a visualization of the solution.
Concepts and Components

Artificial Intelligence

AI is a computer science discipline in which computers are trained to mimic the cognitive functions of the human mind. AI developers train computers to learn and to solve problems in a manner that is similar to, or even superior to, humans. Deep learning and machine learning are subfields of AI. Organizations are increasingly adopting AI, ML, and DL to support their critical business needs. Some examples are as follows:

- Analyzing large amounts of data to unearth previously unknown business insights
- Interacting directly with customers by using natural language processing
- Automating various business processes and functions

Modern AI training and inference workloads require massively parallel computing capabilities. Therefore, GPUs are increasingly being used to execute AI operations because the parallel processing capabilities of GPUs are vastly superior to those of general-purpose CPUs.

Containers

Containers are isolated user-space instances that run on top of a shared host operating system kernel. The adoption of containers is increasing rapidly. Containers offer many of the same application sandboxing benefits that virtual machines (VMs) offer. However, because the hypervisor and guest operating system layers that VMs rely on have been eliminated, containers are far more lightweight. The following figure depicts a visualization of virtual machines versus containers.

Containers also allow the efficient packaging of application dependencies, run times, and so on, directly with an application. The most commonly used container packaging format is the Docker container. An application that has been containerized in the Docker container format can be executed on any machine that can run Docker containers. This is true even if the application’s dependencies are not present on the machine because all dependencies are packaged in the container itself. For more information, visit the Docker website.

Kubernetes

Kubernetes is an open source, distributed, container orchestration platform that was originally designed by Google and is now maintained by the Cloud Native Computing Foundation (CNCF). Kubernetes enables the
automation of deployment, management, and scaling functions for containerized applications. In recent years, Kubernetes has emerged as the dominant container orchestration platform. Although other container packaging formats and run times are supported, Kubernetes is most often used as an orchestration system for Docker containers. For more information, visit the Kubernetes website.

NetApp Trident

Trident is an open source storage orchestrator developed and maintained by NetApp that greatly simplifies the creation, management, and consumption of persistent storage for Kubernetes workloads. Trident, itself a Kubernetes-native application, runs directly within a Kubernetes cluster. With Trident, Kubernetes users (developers, data scientists, Kubernetes administrators, and so on) can create, manage, and interact with persistent storage volumes in the standard Kubernetes format that they are already familiar with. At the same time, they can take advantage of NetApp advanced data management capabilities and a data fabric that is powered by NetApp technology. Trident abstracts away the complexities of persistent storage and makes it simple to consume. For more information, visit the Trident website.

NVIDIA DeepOps

DeepOps is an open source project from NVIDIA that, by using Ansible, automates the deployment of GPU server clusters according to best practices. DeepOps is modular and can be used for various deployment tasks. For this document and the validation exercise that it describes, DeepOps is used to deploy a Kubernetes cluster that consists of GPU server worker nodes. For more information, visit the DeepOps website.

Kubeflow

Kubeflow is an open source AI and ML toolkit for Kubernetes that was originally developed by Google. The Kubeflow project makes deployments of AI and ML workflows on Kubernetes simple, portable, and scalable. Kubeflow abstracts away the intricacies of Kubernetes, allowing data scientists to focus on what they know best—data science. See the following figure for a visualization. Kubeflow has been gaining significant traction as enterprise IT departments have increasingly standardized on Kubernetes. For more information, visit the Kubeflow website.
Kubeflow Pipelines

Kubeflow Pipelines are a key component of Kubeflow. Kubeflow Pipelines are a platform and standard for defining and deploying portable and scalable AI and ML workflows. For more information, see the official Kubeflow documentation.

Jupyter Notebook Server

A Jupyter Notebook Server is an open-source web application that allows data scientists to create wiki-like documents called Jupyter Notebooks that contain live code as well as descriptive text. Jupyter Notebooks are widely used in the AI and ML community as a means of documenting, storing, and sharing AI and ML projects. Kubeflow simplifies the provisioning and deployment of Jupyter Notebook Servers on Kubernetes. For more information on Jupyter Notebooks, visit the Jupyter website. For more information about Jupyter Notebooks within the context of Kubeflow, see the official Kubeflow documentation.

Apache Airflow

Apache Airflow is an open-source workflow management platform that enables programmatic authoring, scheduling, and monitoring for complex enterprise workflows. It is often used to automate ETL and data pipeline workflows, but it is not limited to these types of workflows. The Airflow project was started by Airbnb but has since become very popular in the industry and now falls under the auspices of The Apache Software Foundation. Airflow is written in Python, Airflow workflows are created via Python scripts, and Airflow is
designed under the principle of "configuration as code." Many enterprise Airflow users now run Airflow on top of Kubernetes.

Directed Acyclic Graphs (DAGs)

In Airflow, workflows are called Directed Acyclic Graphs (DAGs). DAGs are made up of tasks that are executed in sequence, in parallel, or a combination of the two, depending on the DAG definition. The Airflow scheduler executes individual tasks on an array of workers, adhering to the task-level dependencies that are specified in the DAG definition. DAGs are defined and created via Python scripts.

NetApp ONTAP 9

NetApp ONTAP 9 is the latest generation of storage management software from NetApp that enables businesses like yours to modernize infrastructure and to transition to a cloud-ready data center. With industry-leading data management capabilities, ONTAP enables you to manage and protect your data with a single set of tools regardless of where that data resides. You can also move data freely to wherever you need it: the edge, the core, or the cloud. ONTAP 9 includes numerous features that simplify data management, accelerate and protect your critical data, and future-proof your infrastructure across hybrid cloud architectures.

Simplify Data Management

Data management is crucial for your enterprise IT operations so that you can use appropriate resources for your applications and datasets. ONTAP includes the following features to streamline and simplify your operations and reduce your total cost of operation:

- **Inline data compaction and expanded deduplication.** Data compaction reduces wasted space inside storage blocks, and deduplication significantly increases effective capacity.
- **Minimum, maximum, and adaptive quality of service (QoS).** Granular QoS controls help maintain performance levels for critical applications in highly shared environments.
- **ONTAP FabricPool.** This feature provides automatic tiering of cold data to public and private cloud storage options, including Amazon Web Services (AWS), Azure, and NetApp StorageGRID object-based storage.

Accelerate and Protect Data

ONTAP delivers superior levels of performance and data protection and extends these capabilities with the following features:

- **High performance and low latency.** ONTAP offers the highest possible throughput at the lowest possible latency.
- **NetApp ONTAP FlexGroup technology.** A FlexGroup volume is a high-performance data container that can scale linearly to up to 20PB and 400 billion files, providing a single namespace that simplifies data management.
- **Data protection.** ONTAP provides built-in data protection capabilities with common management across all platforms.
- **NetApp Volume Encryption.** ONTAP offers native volume-level encryption with both onboard and external key management support.

Future-Proof Infrastructure

ONTAP 9 helps meet your demanding and constantly changing business needs:

- **Seamless scaling and nondisruptive operations.** ONTAP supports the nondisruptive addition of
capacity to existing controllers and to scale-out clusters. You can upgrade to the latest technologies, such as NVMe and 32Gb FC, without costly data migrations or outages.

- **Cloud connection.** ONTAP is one of the most cloud-connected storage management software, with options for software-defined storage (ONTAP Select) and cloud-native instances (NetApp Cloud Volumes Service) in all public clouds.

- **Integration with emerging applications.** By using the same infrastructure that supports existing enterprise apps, ONTAP offers enterprise-grade data services for next-generation platforms and applications such as OpenStack, Hadoop, and MongoDB.

### NetApp Snapshot Copies

A NetApp Snapshot copy is a read-only, point-in-time image of a volume. The image consumes minimal storage space and incurs negligible performance overhead because it only records changes to files create since the last Snapshot copy was made, as depicted in the following figure.

Snapshot copies owe their efficiency to the core ONTAP storage virtualization technology, the Write Anywhere File Layout (WAFL). Like a database, WAFL uses metadata to point to actual data blocks on disk. But, unlike a database, WAFL does not overwrite existing blocks. It writes updated data to a new block and changes the metadata. It's because ONTAP references metadata when it creates a Snapshot copy, rather than copying data blocks, that Snapshot copies are so efficient. Doing so eliminates the seek time that other systems incur in locating the blocks to copy, as well as the cost of making the copy itself.

You can use a Snapshot copy to recover individual files or LUNs or to restore the entire contents of a volume. ONTAP compares pointer information in the Snapshot copy with data on disk to reconstruct the missing or damaged object, without downtime or a significant performance cost.
NetApp FlexClone Technology

NetApp FlexClone technology references Snapshot metadata to create writable, point-in-time copies of a volume. Copies share data blocks with their parents, consuming no storage except what is required for metadata until changes are written to the copy, as depicted in the following figure. Where traditional copies can take minutes or even hours to create, FlexClone software lets you copy even the largest datasets almost instantaneously. That makes it ideal for situations in which you need multiple copies of identical datasets (a development workspace, for example) or temporary copies of a dataset (testing an application against a production dataset).

A Snapshot copy records only changes to the active file system since the last Snapshot copy.
NetApp SnapMirror Data Replication Technology

NetApp SnapMirror software is a cost-effective, easy-to-use unified replication solution across the data fabric. It replicates data at high speeds over LAN or WAN. It gives you high data availability and fast data replication for applications of all types, including business critical applications in both virtual and traditional environments. When you replicate data to one or more NetApp storage systems and continually update the secondary data, your data is kept current and is available whenever you need it. No external replication servers are required. See the following figure for an example of an architecture that leverages SnapMirror technology.

SnapMirror software leverages NetApp ONTAP storage efficiencies by sending only changed blocks over the network. SnapMirror software also uses built-in network compression to accelerate data transfers and reduce network bandwidth utilization by up to 70%. With SnapMirror technology, you can leverage one thin replication data stream to create a single repository that maintains both the active mirror and prior point-in-time copies, reducing network traffic by up to 50%.

FlexClone copies share data blocks with their parents, consuming no storage except what is required for metadata.
NetApp BlueXP Copy and Sync

BlueXP Copy and Sync is a NetApp service for rapid and secure data synchronization. Whether you need to transfer files between on-premises NFS or SMB file shares, NetApp StorageGRID, NetApp ONTAP S3, NetApp Cloud Volumes Service, Azure NetApp Files, AWS S3, AWS EFS, Azure Blob, Google Cloud Storage, or IBM Cloud Object Storage, BlueXP Copy and Sync moves the files where you need them quickly and securely.

After your data is transferred, it is fully available for use on both source and target. BlueXP Copy and Sync can sync data on-demand when an update is triggered or continuously sync data based on a predefined schedule. Regardless, BlueXP Copy and Sync only moves the deltas, so time and money spent on data replication is minimized.

BlueXP Copy and Sync is a software as a service (SaaS) tool that is extremely simple to set up and use. Data transfers that are triggered by BlueXP Copy and Sync are carried out by data brokers. BlueXP Copy and Sync data brokers can be deployed in AWS, Azure, Google Cloud Platform, or on-premises.

NetApp XCP

NetApp XCP is client-based software for any-to-NetApp and NetApp-to-NetApp data migrations and file system insights. XCP is designed to scale and achieve maximum performance by utilizing all available system resources to handle high-volume datasets and high-performance migrations. XCP helps you to gain complete visibility into the file system with the option to generate reports.

NetApp XCP is available in a single package that supports NFS and SMB protocols. XCP includes a Linux binary for NFS data sets and a windows executable for SMB data sets.

NetApp XCP File Analytics is host-based software that detects file shares, runs scans on the file system, and provides a dashboard for file analytics. XCP File Analytics is compatible with both NetApp and non-NetApp systems and runs on Linux or Windows hosts to provide analytics for NFS and SMB-exported file systems.

NetApp ONTAP FlexGroup Volumes

A training dataset can be a collection of potentially billions of files. Files can include text, audio, video, and other forms of unstructured data that must be stored and processed to be read in parallel. The storage system
must store large numbers of small files and must read those files in parallel for sequential and random I/O.

A FlexGroup volume is a single namespace that comprises multiple constituent member volumes, as shown in the following figure. From a storage administrator viewpoint, a FlexGroup volume is managed and acts like a NetApp FlexVol volume. Files in a FlexGroup volume are allocated to individual member volumes and are not striped across volumes or nodes. They enable the following capabilities:

- FlexGroup volumes provide multiple petabytes of capacity and predictable low latency for high-metadata workloads.
- They support up to 400 billion files in the same namespace.
- They support parallelized operations in NAS workloads across CPUs, nodes, aggregates, and constituent FlexVol volumes.

Hardware and Software Requirements

The NetApp AI Control Plane solution is not dependent on this specific hardware. The solution is compatible with any NetApp physical storage appliance, software-defined instance, or cloud service, that is supported by Trident. Examples include a NetApp AFF storage system, Azure NetApp Files, NetApp Cloud Volumes Service, a NetApp ONTAP Select software-defined storage instance, or a NetApp Cloud Volumes ONTAP instance. Additionally, the solution can be implemented on any Kubernetes cluster as long as the Kubernetes version used is supported by Kubeflow and NetApp Trident. For a list of Kubernetes versions that are supported by Kubeflow, see the [official Kubeflow documentation](#). For a list of Kubernetes versions that are supported by Trident, see the [Trident documentation](#). See the following tables for details on the environment that was used to validate the solution.

<table>
<thead>
<tr>
<th>Infrastructure Component</th>
<th>Quantity</th>
<th>Details</th>
<th>Operating System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment jump host</td>
<td>1</td>
<td>VM</td>
<td>Ubuntu 20.04.2 LTS</td>
</tr>
<tr>
<td>Kubernetes master nodes</td>
<td>1</td>
<td>VM</td>
<td>Ubuntu 20.04.2 LTS</td>
</tr>
</tbody>
</table>
Infrastructure Component | Quantity | Details | Operating System
--- | --- | --- | ---
Kubernetes worker nodes | 2 | VM | Ubuntu 20.04.2 LTS
Kubernetes GPU worker nodes | 2 | NVIDIA DGX-1 (bare-metal) | NVIDIA DGX OS 4.0.5 (based on Ubuntu 18.04.2 LTS)
Storage | 1 HA Pair | NetApp AFF A220 | NetApp ONTAP 9.7 P6

Software Component | Version
--- | ---
Apache Airflow | 2.0.1
Apache Airflow Helm Chart | 8.0.8
Docker | 19.03.12
Kubeflow | 1.2
Kubernetes | 1.18.9
NetApp Trident | 21.01.2
NVIDIA DeepOps | Trident deployment functionality from master branch as of commit 61898cdfda; All other functionality from version 21.03

Support

NetApp does not offer enterprise support for Apache Airflow, Docker, Kubeflow, Kubernetes, or NVIDIA DeepOps. If you are interested in a fully supported solution with capabilities similar to the NetApp AI Control Plane solution, contact NetApp about fully supported AI/ML solutions that NetApp offers jointly with partners.

Kubernetes Deployment

This section describes the tasks that you must complete to deploy a Kubernetes cluster in which to implement the NetApp AI Control Plane solution. If you already have a Kubernetes cluster, then you can skip this section as long as you are running a version of Kubernetes that is supported by Kubeflow and NetApp Trident. For a list of Kubernetes versions that are supported by Kubeflow, see the official Kubeflow documentation. For a list of Kubernetes versions that are supported by Trident, see the Trident documentation.

For on-premises Kubernetes deployments that incorporate bare-metal nodes featuring NVIDIA GPU(s), NetApp recommends using NVIDIA’s DeepOps Kubernetes deployment tool. This section outlines the deployment of a Kubernetes cluster using DeepOps.

Prerequisites

Before you perform the deployment exercise that is outlined in this section, we assume that you have already performed the following tasks:

1. You have already configured any bare-metal Kubernetes nodes (for example, an NVIDIA DGX system that is part of an ONTAP AI pod) according to standard configuration instructions.
2. You have installed a supported operating system on all Kubernetes master and worker nodes and on a
deployment jump host. For a list of operating systems that are supported by DeepOps, see the DeepOps
GitHub site.

Use NVIDIA DeepOps to Install and Configure Kubernetes

To deploy and configure your Kubernetes cluster with NVIDIA DeepOps, perform the following tasks from a
deployment jump host:

1. Download NVIDIA DeepOps by following the instructions on the Getting Started page on the NVIDIA
   DeepOps GitHub site.
2. Deploy Kubernetes in your cluster by following the instructions on the Kubernetes Deployment Guide page
   on the NVIDIA DeepOps GitHub site.

NetApp Trident Deployment and Configuration

NetApp Trident Deployment and Configuration

This section describes the tasks that you must complete to install and configure NetApp Trident in your Kubernetes cluster.

Prerequisites

Before you perform the deployment exercise that is outlined in this section, we assume that you have already
performed the following tasks:

1. You already have a working Kubernetes cluster, and you are running a version of Kubernetes that is
   supported by Trident. For a list of supported versions, see the Trident documentation.
2. You already have a working NetApp storage appliance, software-defined instance, or cloud storage
   service, that is supported by Trident.

Install Trident

To install and configure NetApp Trident in your Kubernetes cluster, perform the following tasks from the
deployment jump host:

1. Deploy Trident using one of the following methods:
   - If you used NVIDIA DeepOps to deploy your Kubernetes cluster, you can also use NVIDIA DeepOps to
deploy Trident in your Kubernetes cluster. To deploy Trident with DeepOps, follow the Trident
deployment instructions on the NVIDIA DeepOps GitHub site.
   - If you did not use NVIDIA DeepOps to deploy your Kubernetes cluster or if you simply prefer to deploy
   Trident manually, you can deploy Trident by following the deployment instructions in the Trident
documentation. Be sure to create at least one Trident Backend and at least one Kubernetes
   StorageClass, for more information about how to configure Backends and StorageClasses see the
   linked subsections at NetApp Docs.

   If you are deploying the NetApp AI Control Plane solution on an ONTAP AI pod, see
   Example Trident Backends for ONTAP AI Deployments for some examples of different
   Trident Backends that you might want to create and Example Kubernetes
   Storageclasses for ONTAP AI Deployments for some examples of different Kubernetes
   StorageClasses that you might want to create.
Example Trident Backends for ONTAP AI Deployments

Before you can use Trident to dynamically provision storage resources within your Kubernetes cluster, you must create one or more Trident Backends. The examples that follow represent different types of Backends that you might want to create if you are deploying the NetApp AI Control Plane solution on an ONTAP AI pod. For more information about Backends, see the Trident documentation.

1. NetApp recommends creating a FlexGroup-enabled Trident Backend for each data LIF (logical network interface that provides data access) that you want to use on your NetApp AFF system. This will allow you to balance volume mounts across LIFs.

The example commands that follow show the creation of two FlexGroup-enabled Trident Backends for two different data LIFs that are associated with the same ONTAP storage virtual machine (SVM). These Backends use the `ontap-nas-flexgroup` storage driver. ONTAP supports two main data volume types: FlexVol and FlexGroup. FlexVol volumes are size-limited (as of this writing, the maximum size depends on the specific deployment). FlexGroup volumes, on the other hand, can scale linearly to up to 20PB and 400 billion files, providing a single namespace that greatly simplifies data management. Therefore, FlexGroup volumes are optimal for AI and ML workloads that rely on large amounts of data.

If you are working with a small amount of data and want to use FlexVol volumes instead of FlexGroup volumes, you can create Trident Backends that use the `ontap-nas` storage driver instead of the `ontap-nas-flexgroup` storage driver.

```
$ cat << EOF > ./trident-backend-ontap-ai-flexgroups-iface1.json
{
  "version": 1,
  "storageDriverName": "ontap-nas-flexgroup",
  "backendName": "ontap-ai-flexgroups-iface1",
  "managementLIF": "10.61.218.100",
  "dataLIF": "192.168.11.11",
  "svm": "ontapai_nfs",
  "username": "admin",
  "password": "ontapai"
}
EOF

$ tridentctl create backend -f ./trident-backend-ontap-ai-flexgroups-iface1.json -n trident
```

<table>
<thead>
<tr>
<th>UUID</th>
<th>NAME</th>
<th>STORAGE DRIVER</th>
<th>STATE</th>
<th>VOLUMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>b74cbdddb-e0b8-40b7-b263-b6da6dec0bdd</td>
<td>ontap-ai-flexgroups-iface1</td>
<td>ontap-nas-flexgroup</td>
<td>online</td>
<td>0</td>
</tr>
</tbody>
</table>
2. NetApp also recommends creating one or more FlexVol-enabled Trident Backends. If you use FlexGroup volumes for training dataset storage, you might want to use FlexVol volumes for storing results, output, debug information, and so on. If you want to use FlexVol volumes, you must create one or more FlexVol-enabled Trident Backends. The example commands that follow show the creation of a single FlexVol-enabled Trident Backend that uses a single data LIF.
Example Kubernetes StorageClasses for ONTAP AI Deployments

Before you can use Trident to dynamically provision storage resources within your Kubernetes cluster, you must create one or more Kubernetes StorageClasses. The examples that follow represent different types of StorageClasses that you might want to
create if you are deploying the NetApp AI Control Plane solution on an ONTAP AI pod. For more information about StorageClasses, see the Trident documentation.

1. NetApp recommends creating a separate StorageClass for each FlexGroup-enabled Trident Backend that you created in the section Example Trident Backends for ONTAP AI Deployments, step 1. These granular StorageClasses enable you to add NFS mounts that correspond to specific LIFs (the LIFs that you specified when you created the Trident Backends) as a particular Backend that is specified in the StorageClass spec file. The example commands that follow show the creation of two StorageClasses that correspond to the two example Backends that were created in the section Example Trident Backends for ONTAP AI Deployments, step 1. For more information about StorageClasses, see the Trident documentation.

So that a persistent volume isn’t deleted when the corresponding PersistentVolumeClaim (PVC) is deleted, the following example uses a reclaimPolicy value of Retain. For more information about the reclaimPolicy field, see the official Kubernetes documentation.
2. NetApp also recommends creating a StorageClass that corresponds to the FlexVol-enabled Trident Backend that you created in the section Example Trident Backends for ONTAP AI Deployments, step 2. The example commands that follow show the creation of a single StorageClass for FlexVol volumes.

In the following example, a particular Backend is not specified in the StorageClass definition file because only one FlexVol-enabled Trident backend was created. When you use Kubernetes to administer volumes that use this StorageClass, Trident attempts to use any available backend that uses the ontap-nas driver.
3. NetApp also recommends creating a generic StorageClass for FlexGroup volumes. The following example commands show the creation of a single generic StorageClass for FlexGroup volumes.

Note that a particular backend is not specified in the StorageClass definition file. Therefore, when you use Kubernetes to administer volumes that use this StorageClass, Trident attempts to use any available backend that uses the `ontap-nas-flexgroup` driver.
Kubeflow Deployment

This section describes the tasks that you must complete to deploy Kubeflow in your Kubernetes cluster.

Prerequisites

Before you perform the deployment exercise that is outlined in this section, we assume that you have already performed the following tasks:

1. You already have a working Kubernetes cluster, and you are running a version of Kubernetes that is supported by Kubeflow. For a list of supported versions, see the official Kubeflow documentation.
2. You have already installed and configured NetApp Trident in your Kubernetes cluster as outlined in Trident Deployment and Configuration.

Set Default Kubernetes StorageClass

Before you deploy Kubeflow, you must designate a default StorageClass within your Kubernetes cluster. The Kubeflow deployment process attempts to provision new persistent volumes using the default StorageClass. If no StorageClass is designated as the default StorageClass, then the deployment fails. To designate a default StorageClass within your cluster, perform the following task from the deployment jump host. If you have already designated a default StorageClass within your cluster, then you can skip this step.

1. Designate one of your existing StorageClasses as the default StorageClass. The example commands that follow show the designation of a StorageClass named ontap-ai- flexvols-retain as the default StorageClass.

   The ontap-nas-flexgroup Trident Backend type has a minimum PVC size that is fairly large. By default, Kubeflow attempts to provision PVCs that are only a few GBs in size. Therefore, you should not designate a StorageClass that utilizes the ontap-nas-flexgroup Backend type as the default StorageClass for the purposes of Kubeflow deployment.

   ```
   $ kubectl get sc
   NAME                                PROVISIONER             AGE
   ontap-ai-flexgroups-retain          csi.trident.netapp.io   25h
   ontap-ai-flexgroups-retain-iface1   csi.trident.netapp.io   25h
   ontap-ai-flexgroups-retain-iface2   csi.trident.netapp.io   25h
   ontap-ai-flexvols-retain            csi.trident.netapp.io   3s
   $ kubectl patch storageclass ontap-ai-flexvols-retain -p '{"metadata":
   "annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}'
   storageclass.storage.k8s.io/ontap-ai-flexvols-retain patched
   $ kubectl get sc
   NAME                                PROVISIONER             AGE
   ontap-ai-flexgroups-retain          csi.trident.netapp.io   25h
   ontap-ai-flexgroups-retain-iface1   csi.trident.netapp.io   25h
   ontap-ai-flexgroups-retain-iface2   csi.trident.netapp.io   25h
   ontap-ai-flexvols-retain (default)  csi.trident.netapp.io   54s
   ```
Use NVIDIA DeepOps to Deploy Kubeflow

NetApp recommends using the Kubeflow deployment tool that is provided by NVIDIA DeepOps. To deploy Kubeflow in your Kubernetes cluster using the DeepOps deployment tool, perform the following tasks from the deployment jump host.

1. Deploy Kubeflow in your cluster by following the Kubeflow deployment instructions on the NVIDIA DeepOps GitHub site.

2. Note down the Kubeflow Dashboard URL that the DeepOps Kubeflow deployment tool outputs.

```
$ ./scripts/k8s/deploy_kubeflow.sh -x

INFO[0007] Applied the configuration Successfully!
filename="cmd/apply.go:72"
Kubeflow app installed to: /home/ai/kubeflow
It may take several minutes for all services to start. Run 'kubectl get pods -n kubeflow' to verify
To remove (excluding CRDs, istio, auth, and cert-manager), run: ./scripts/k8s_deploy_kubeflow.sh -d
To perform a full uninstall : ./scripts/k8s_deploy_kubeflow.sh -D
Kubeflow Dashboard (HTTP NodePort): http://10.61.188.111:31380
```

3. Confirm that all pods deployed within the Kubeflow namespace show a STATUS of Running and confirm that no components deployed within the namespace are in an error state. It may take several minutes for all pods to start.

```
$ kubectl get all -n kubeflow
NAME                                                                   READY
STATUS  RESTARTS   AGE
pod/admission-webhook-bootstrap-stateful-set-0                     1/1
Running  0          95s
pod/admission-webhook-deployment-6b89c84c98-vrtbh                    1/1
Running  0          91s
pod/application-controller-stateful-set-0                          1/1
Running  0          98s
pod/argo-ui-5dfc5d8b4f-m2wn4                                       1/1
Running  0          97s
pod/centraldashboard-cf4874ddc-7hcr8                                1/1
Running  0          97s
pod/jupyter-web-app-deployment-685b455447-gjhh7                      1/1
Running  0          96s
pod/katib-controller-88c97d85c-kgg66                                 1/1
Running  1          95s
```
pod/katib-db-8598468fd8-5jw2c 1/1
Running 0 95s
pod/katib-manager-574c8c67f9-wtrf5 1/1
Running 1 95s
pod/katib-manager-rest-778857c989-fjbzn 1/1
Running 0 95s
pod/katib-suggestion-bayesianoptimization-65df4d7455-qthmw 1/1
Running 0 94s
pod/katib-suggestion-grid-56bf69f597-98vwn 1/1
Running 0 94s
pod/katib-suggestion-hyperband-7777b76cb9-9v6dq 1/1
Running 0 93s
pod/katib-suggestion-nasrl-77f6f9458c-2qzxq 1/1
Running 0 93s
pod/katib-suggestion-random-77b88b5c79-164j9 1/1
Running 0 93s
pod/katib-ui-7587c5b967-nd629 1/1
Running 0 95s
pod/metacontroller-0 1/1
Running 0 96s
pod/metadata-db-5dd459cc-swzkm 1/1
Running 0 94s
pod/metadata-deployment-6cf77db994-69fk7 1/1
Running 3 93s
pod/metadata-deployment-6cf77db994-mpbjt 1/1
Running 3 93s
pod/metadata-deployment-6cf77db994-xg7tz 1/1
Running 3 94s
pod/metadata-ui-78f5b59b56-qb6kr 1/1
Running 0 94s
pod/minio-758b769d67-llvdr 1/1
Running 0 91s
pod/ml-pipeline-5875b9db95-g8t2k 1/1
Running 0 91s
pod/ml-pipeline-persistenceagent-9b69ddd46-bt9r9 1/1
Running 0 90s
pod/ml-pipeline-scheduledworkflow-7b8d756c76-7x56s 1/1
Running 0 90s
pod/ml-pipeline-ui-79ffd9c76-fcwpd 1/1
Running 0 90s
pod/ml-pipeline-viewer-controller-deployment-5fdc87f58-b2t9r 1/1
Running 0 90s
pod/mysql-657f87857d-15k9z 1/1
Running 0 91s
pod/notebook-controller-deployment-56b4f59bbf-8bvn 1/1
Running 0 92s
pod/profiles-deployment-6bc745947-mrdkh
Running 0 90s
pod/pytorch-operator-77c97f4879-hmlrv
Running 0 92s
pod/seldon-operator-controller-manager-0
Running 1 91s
pod/spartakus-volunteer-5fdffdd779-17qkm
Running 0 92s
pod/tensorboard-6544748d94-nh8b2
Running 0 92s
pod/tf-job-dashboard-56f79c59dd-6w59t
Running 0 92s
pod/tf-job-operator-79cbfd6dbc-rb58c
Running 0 91s
pod/workflow-controller-db644d554-cwrnb
Running 0 97s

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLUSTER-IP</th>
<th>EXTERNAL-IP</th>
<th>PORT(S)</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>service/admission-webhook-service</td>
<td>ClusterIP</td>
<td>10.233.51.169</td>
<td>&lt;none&gt;</td>
<td>443/TCP</td>
<td>97s</td>
</tr>
<tr>
<td>service/application-controller-service</td>
<td>ClusterIP</td>
<td>10.233.4.54</td>
<td>&lt;none&gt;</td>
<td>443/TCP</td>
<td>98s</td>
</tr>
<tr>
<td>service/argo-ui</td>
<td>NodePort</td>
<td>10.233.47.191</td>
<td>&lt;none&gt;</td>
<td>80:31799/TCP</td>
<td>97s</td>
</tr>
<tr>
<td>service/centraldashboard</td>
<td>ClusterIP</td>
<td>10.233.8.36</td>
<td>&lt;none&gt;</td>
<td>80/TCP</td>
<td>97s</td>
</tr>
<tr>
<td>service/jupyter-web-app-service</td>
<td>ClusterIP</td>
<td>10.233.1.42</td>
<td>&lt;none&gt;</td>
<td>80/TCP</td>
<td>97s</td>
</tr>
<tr>
<td>service/katib-controller</td>
<td>ClusterIP</td>
<td>10.233.25.226</td>
<td>&lt;none&gt;</td>
<td>443/TCP</td>
<td>96s</td>
</tr>
<tr>
<td>service/katib-db</td>
<td>ClusterIP</td>
<td>10.233.33.151</td>
<td>&lt;none&gt;</td>
<td>3306/TCP</td>
<td>97s</td>
</tr>
<tr>
<td>service/katib-manager</td>
<td>ClusterIP</td>
<td>10.233.46.239</td>
<td>&lt;none&gt;</td>
<td>6789/TCP</td>
<td>96s</td>
</tr>
<tr>
<td>service/katib-manager-rest</td>
<td>ClusterIP</td>
<td>10.233.55.32</td>
<td>&lt;none&gt;</td>
<td>80/TCP</td>
<td>96s</td>
</tr>
<tr>
<td>service/katib-suggestion-bayesianoptimization</td>
<td>ClusterIP</td>
<td>10.233.49.191</td>
<td>&lt;none&gt;</td>
<td>6789/TCP</td>
<td>95s</td>
</tr>
<tr>
<td>service/katib-suggestion-grid</td>
<td>ClusterIP</td>
<td>10.233.9.105</td>
<td>&lt;none&gt;</td>
<td>6789/TCP</td>
<td>95s</td>
</tr>
<tr>
<td>service/katib-suggestion-hyperband</td>
<td>ClusterIP</td>
<td>10.233.22.2</td>
<td>&lt;none&gt;</td>
<td>6789/TCP</td>
<td>95s</td>
</tr>
<tr>
<td>service/katib-suggestion-nasrl</td>
<td>ClusterIP</td>
<td>10.233.63.73</td>
<td>&lt;none&gt;</td>
<td>6789/TCP</td>
<td>95s</td>
</tr>
<tr>
<td>service/katib-suggestion-random</td>
<td>ClusterIP</td>
<td>10.233.57.210</td>
<td>&lt;none&gt;</td>
<td>6789/TCP</td>
<td>95s</td>
</tr>
<tr>
<td>SERVICE NAME</td>
<td>CLUSTERIP ADDRESS</td>
<td>PORT/PROTO</td>
<td>AGE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------</td>
<td>------------</td>
<td>-------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/katib-ui</td>
<td>10.233.6.116</td>
<td>80/TCP</td>
<td>96s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/metadata-db</td>
<td>10.233.31.2</td>
<td>3306/TCP</td>
<td>96s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/metadata-service</td>
<td>10.233.27.104</td>
<td>8080/TCP</td>
<td>96s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/metadata-ui</td>
<td>10.233.57.177</td>
<td>80/TCP</td>
<td>96s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/minio-service</td>
<td>10.233.44.90</td>
<td>9000/TCP</td>
<td>94s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/ml-pipeline</td>
<td>10.233.41.201</td>
<td>8888/TCP,8887/TCP</td>
<td>94s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/ml-pipeline-service</td>
<td>10.233.36.207</td>
<td>80/TCP</td>
<td>93s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/ml-pipeline-ui</td>
<td>10.233.61.150</td>
<td>80/TCP</td>
<td>93s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/mysql</td>
<td>10.233.55.117</td>
<td>3306/TCP</td>
<td>94s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/notebook-controller-service</td>
<td>10.233.10.166</td>
<td>443/TCP</td>
<td>95s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/profiles-kfam</td>
<td>10.233.33.79</td>
<td>8081/TCP</td>
<td>92s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/pytorch-operator</td>
<td>10.233.37.112</td>
<td>8443/TCP</td>
<td>95s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/seldon-operator-controller-manager-service</td>
<td>10.233.30.178</td>
<td>443/TCP</td>
<td>92s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/tensorboard</td>
<td>10.233.58.151</td>
<td>9000/TCP</td>
<td>94s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/tf-job-dashboard</td>
<td>10.233.4.17</td>
<td>80/TCP</td>
<td>94s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/tf-job-operator</td>
<td>10.233.60.32</td>
<td>8443/TCP</td>
<td>94s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service/webhook-server-service</td>
<td>10.233.32.167</td>
<td>443/TCP</td>
<td>87s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>deployment.apps/admission-webhook-deployment</td>
<td>1/1</td>
<td>1</td>
</tr>
<tr>
<td>deployment.apps/argo-ui</td>
<td>1/1</td>
<td>1</td>
</tr>
<tr>
<td>deployment.apps/centraldashboard</td>
<td>1/1</td>
<td>1</td>
</tr>
<tr>
<td>deployment.apps/jupyter-web-app-deployment</td>
<td>1/1</td>
<td>1</td>
</tr>
<tr>
<td>deployment.apps/katib-controller</td>
<td>1/1</td>
<td>1</td>
</tr>
</tbody>
</table>

NAME TO-DATE AVAILABLE AGE

98
<table>
<thead>
<tr>
<th>Deployment</th>
<th>Replicas</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>deployment.apps/katib-db</td>
<td>1/1</td>
<td>97s</td>
</tr>
<tr>
<td>deployment.apps/katib-manager</td>
<td>1/1</td>
<td>96s</td>
</tr>
<tr>
<td>deployment.apps/katib-manager-rest</td>
<td>1/1</td>
<td>96s</td>
</tr>
<tr>
<td>deployment.apps/katib-suggestion-bayesianoptimization</td>
<td>1/1</td>
<td>95s</td>
</tr>
<tr>
<td>deployment.apps/katib-suggestion-grid</td>
<td>1/1</td>
<td>95s</td>
</tr>
<tr>
<td>deployment.apps/katib-suggestion-hyperband</td>
<td>1/1</td>
<td>95s</td>
</tr>
<tr>
<td>deployment.apps/katib-suggestion-nasrl</td>
<td>1/1</td>
<td>95s</td>
</tr>
<tr>
<td>deployment.apps/katib-suggestion-random</td>
<td>1/1</td>
<td>95s</td>
</tr>
<tr>
<td>deployment.apps/katib-ui</td>
<td>1/1</td>
<td>96s</td>
</tr>
<tr>
<td>deployment.apps/metadata-db</td>
<td>1/1</td>
<td>96s</td>
</tr>
<tr>
<td>deployment.apps/metadata-deployment</td>
<td>3/3</td>
<td>96s</td>
</tr>
<tr>
<td>deployment.apps/metadata-ui</td>
<td>1/1</td>
<td>96s</td>
</tr>
<tr>
<td>deployment.apps/minio</td>
<td>1/1</td>
<td>94s</td>
</tr>
<tr>
<td>deployment.apps/ml-pipeline</td>
<td>1/1</td>
<td>94s</td>
</tr>
<tr>
<td>deployment.apps/ml-pipeline-persistenceagent</td>
<td>1/1</td>
<td>93s</td>
</tr>
<tr>
<td>deployment.apps/ml-pipeline-scheduledworkflow</td>
<td>1/1</td>
<td>93s</td>
</tr>
<tr>
<td>deployment.apps/ml-pipeline-ui</td>
<td>1/1</td>
<td>93s</td>
</tr>
<tr>
<td>deployment.apps/ml-pipeline-viewer-controller-deployment</td>
<td>1/1</td>
<td>93s</td>
</tr>
<tr>
<td>deployment.apps/mysql</td>
<td>1/1</td>
<td>94s</td>
</tr>
<tr>
<td>deployment.apps/notebook-controller-deployment</td>
<td>1/1</td>
<td>95s</td>
</tr>
<tr>
<td>deployment.apps/profiles-deployment</td>
<td>1/1</td>
<td>92s</td>
</tr>
<tr>
<td>deployment.apps/pytorch-operator</td>
<td>1/1</td>
<td>95s</td>
</tr>
<tr>
<td>deployment.apps/spartakus-volunteer</td>
<td>1/1</td>
<td>94s</td>
</tr>
<tr>
<td>NAME</td>
<td>DESIRED</td>
<td>CURRENT</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>deployment.apps/tensorboard</td>
<td>1/1</td>
<td>1/1</td>
</tr>
<tr>
<td>deployment.apps/tf-job-dashboard</td>
<td>1/1</td>
<td>1/1</td>
</tr>
<tr>
<td>deployment.apps/tf-job-operator</td>
<td>1/1</td>
<td>1/1</td>
</tr>
<tr>
<td>deployment.apps/workflow-controller</td>
<td>1/1</td>
<td>1/1</td>
</tr>
<tr>
<td>deployment.apps/admission-webhook-deployment-6b89c84c98</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/argo-ui-5dcf5d8b4f</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/centraldashboard-cf4874ddc</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/jupyter-web-app-deployment-685b455447</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/katib-controller-88c97d85c</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/katib-db-8598468fd8</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/katib-manager-574c8c67f9</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/katib-manager-rest-778857c989</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/katib-suggestion-bayesianoptimization-65df4d7455</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/katib-suggestion-grid-56bf69f597</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/katib-suggestion-hyperband-7777b76cb9</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/katib-suggestion-nasrl-77f6f9458c</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/katib-suggestion-random-77b88b5c79</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/katib-ui-7587c5b967</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/metadata-db-5dd459cc</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/metadata-deployment-6cf77db994</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>replicaset.apps/metadata-ui-78f5b59b56</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>replicaset.apps/minio-758b769d67</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
replicaset.apps/ml-pipeline-5875b9db95  1
replicaset.apps/ml-pipeline-persistenceagent-9b69d5d46  1
replicaset.apps/ml-pipeline-scheduledworkflow-7b8d756c76  1
replicaset.apps/ml-pipeline-ui-79f9d97c76  1
replicaset.apps/ml-pipeline-viewer-controller-deployment-5fdd87f58  1
replicaset.apps/mysql-657f87857d  1
replicaset.apps/notebook-controller-deployment-56b4f59bbf  1
replicaset.apps/profiles-deployment-6bc745947  1
replicaset.apps/pytorch-operator-77c97f4879  1
replicaset.apps/spartakus-volunteer-5fdddb779  1
replicaset.apps/tensorboard-6547758d94  1
replicaset.apps/tf-job-dashboard-56f79c59dd  1
replicaset.apps/tf-job-operator-79cbfd6dbc  1
replicaset.apps/workflow-controller-db644d554  1

NAME                                                        READY   AGE
statefulset.apps/admission-webhook-bootstrap-stateful-set   1/1     97s
statefulset.apps/application-controller-stateful-set        1/1     98s
statefulset.apps/metacontroller                             1/1     98s
statefulset.apps/seldon-operator-controller-manager         1/1     98s

$ kubectl get pvc -n kubeflow

NAME             STATUS   VOLUME                                      CAPACITY   ACCESS MODES   STORAGECLASS               AGE
katib-mysql      Bound    pvc-b07f293e-d028-11e9-9b9d-00505681a82d 10Gi       RWO            ontap-ai-flexvols-retain  27m
metadata-mysql    Bound    pvc-b0f3f32e-d028-11e9-9b9d-00505681a82d 10Gi       RWO            ontap-ai-flexvols-retain  27m
minio-pv-claim    Bound    pvc-b22727ee-d028-11e9-9b9d-00505681a82d 20Gi       RWO            ontap-ai-flexvols-retain  27m
mysql-pv-claim    Bound    pvc-b2429af5-d028-11e9-9b9d-00505681a82d 20Gi       RWO            ontap-ai-flexvols-retain  27m

4. In your web browser, access the Kubeflow central dashboard by navigating to the URL that you noted
down in step 2.

The default username is admin@kubeflow.org, and the default password is 12341234. To create additional users, follow the instructions in the official Kubeflow documentation.

---

Example Kubeflow Operations and Tasks

This section includes examples of various operations and tasks that you may want to perform using Kubeflow.

---

Provision a Jupyter Notebook Workspace for Data Scientist or Developer Use

Kubeflow is capable of rapidly provisioning new Jupyter Notebook servers to act as data scientist workspaces. To provision a new Jupyter Notebook server with Kubeflow, perform the following tasks. For more information about Jupyter Notebooks within the Kubeflow
context, see the official Kubeflow documentation.

1. From the Kubeflow central dashboard, click Notebook Servers in the main menu to navigate to the Jupyter Notebook server administration page.

2. Click New Server to provision a new Jupyter Notebook server.
3. Give your new server a name, choose the Docker image that you want your server to be based on, and specify the amount of CPU and RAM to be reserved by your server. If the Namespace field is blank, use the Select Namespace menu in the page header to choose a namespace. The Namespace field is then auto-populated with the chosen namespace.

In the following example, the `kubeflow-anonymous` namespace is chosen. In addition, the default values for Docker image, CPU, and RAM are accepted.
4. Specify the workspace volume details. If you choose to create a new volume, then that volume or PVC is provisioned using the default StorageClass. Because a StorageClass utilizing Trident was designated as the default StorageClass in the section **Kubeflow Deployment**, the volume or PVC is provisioned with Trident. This volume is automatically mounted as the default workspace within the Jupyter Notebook Server container. Any notebooks that a user creates on the server that are not saved to a separate data volume are automatically saved to this workspace volume. Therefore, the notebooks are persistent across reboots.

5. Add data volumes. The following example specifies an existing PVC named 'pb-fg-all' and accepts the default mount point.
6. **Optional**: Request that the desired number of GPUs be allocated to your notebook server. In the following example, one GPU is requested.

7. Click Launch to provision your new notebook server.

8. Wait for your notebook server to be fully provisioned. This can take several minutes if you have never provisioned a server using the Docker image that you specified because the image needs to be downloaded. When your server has been fully provisioned, you see a green check mark in the Status column on the Jupyter Notebook server administration page.
9. Click Connect to connect to your new server web interface.

10. Confirm that the dataset volume that was specified in step 6 is mounted on the server. Note that this volume is mounted within the default workspace by default. From the perspective of the user, this is just another folder within the workspace. The user, who is likely a data scientist and not an infrastructure expert, does not need to possess any storage expertise in order to use this volume.
11. Open a Terminal and, assuming that a new volume was requested in step 5, execute `df -h` to confirm that a new Trident-provisioned persistent volume is mounted as the default workspace.

The default workspace directory is the base directory that you are presented with when you first access the server’s web interface. Therefore, any artifacts that you create by using the web interface are stored on this Trident-provisioned persistent volume.
12. Using the terminal, run `nvidia-smi` to confirm that the correct number of GPUs were allocated to the notebook server. In the following example, one GPU has been allocated to the notebook server as requested in step 7.

```bash
$ df -h
Filesystem Size Used Avail
overlay 439G 34G 382G
tmpfs 64M 0 64M
0% /dev
tmpfs 252G 0 252G
0% /sys/fs/cgroup
/dev/sda2 439G 34G 382G
9% /etc/hosts
192.168.11.111:/trident_pvc_3dcfe7e5_d5a9_11e9_9b9d_00505681a882d 10G 320K 10G
1% /home/jovyan
tmpfs 252G 0 252G
0% /dev/shm
192.168.11.111:/pb fg_all 10T 10T 47G
100% /home/jovyan/data-vol-1
tmpfs 252G 12K 252G
1% /run/secrets/kubernetes.io/serviceaccount
tmpfs 252G 12K 252G
1% /proc/driver/nvidia
tmpfs 51G 4.9M 51G
1% /run/nvidia-persistented/socket
udev 252G 0 252G
0% /dev/nvidia5
tmpfs 252G 0 252G
0% /proc/acpi
tmpfs 252G 0 252G
0% /proc/scsi
tmpfs 252G 0 252G
0% /sys/firmware

$ nvidia-smi
Fri Sep 13 13:52:15 2019
NVIDIA-SMI 410.104 Driver Version: 410.104 CUDA Version: N/A
-------------------------------
| GPU Name | Persistence-M | Bus-Id | Memory-Usage | Volatile Uncorr. ECC |
| Fan Temp | Perf Pwr:Usage/Cap | Memory-Usage | GPU-Util Compute M. |
-------------------------------
| Tesla V100-SXM2... On | | 00000.000.000:00.0 Off |
| N/A | 38C PO 46W / 300W |
-------------------------------
| Processes: |
| GPU | PID Type Process name GPU Memory Usage |
| GPU | 0% Default |

No running processes found

$
Example Notebooks and Pipelines

The NetApp Data Science Toolkit for Kubernetes can be used in conjunction with Kubeflow. Using the NetApp Data Science Toolkit with Kubeflow provides the following benefits:

- Data scientists can perform advanced NetApp data management operations directly from within a Jupyter Notebook.
- Advanced NetApp data management operations can be incorporated into automated workflows using the Kubeflow Pipelines framework.

Refer to the Kubeflow Examples section within the NetApp Data Science Toolkit GitHub repository for details on using the toolkit with Kubeflow.

Apache Airflow Deployment

NetApp recommends running Apache Airflow on top of Kubernetes. This section describes the tasks that you must complete to deploy Airflow in your Kubernetes cluster.

Prerequisites

Before you perform the deployment exercise that is outlined in this section, we assume that you have already performed the following tasks:

1. You already have a working Kubernetes cluster.
2. You have already installed and configured NetApp Trident in your Kubernetes cluster as outlined in the section “NetApp Trident Deployment and Configuration.”

Install Helm

Airflow is deployed using Helm, a popular package manager for Kubernetes. Before you deploy Airflow, you must install Helm on the deployment jump host. To install Helm on the deployment jump host, follow the installation instructions in the official Helm documentation.

Set Default Kubernetes StorageClass

Before you deploy Airflow, you must designate a default StorageClass within your Kubernetes cluster. The Airflow deployment process attempts to provision new persistent volumes using the default StorageClass. If no StorageClass is designated as the default StorageClass, then the deployment fails. To designate a default StorageClass within your cluster, follow the instructions outlined in the section Kubeflow Deployment. If you have already designated a default StorageClass within your cluster, then you can skip this step.

Use Helm to Deploy Airflow

To deploy Airflow in your Kubernetes cluster using Helm, perform the following tasks from the deployment jump host:

1. Deploy Airflow using Helm by following the deployment instructions for the official Airflow chart on the Artifact Hub. The example commands that follow show the deployment of Airflow using Helm. Modify, add, and/or remove values in the custom-values.yaml file as needed depending on your environment and
desired configuration.

$ cat << EOF > custom-values.yaml
#########################################
# Airflow - Common Configs
#########################################
airflow:
  ## the airflow executor type to use
  executor: "CeleryExecutor"
  ## environment variables for the web/scheduler/worker Pods (for
  airflow configs)
  
  #
  #
  # Airflow - WebUI Configs
  # Airflow - Logs Configs
  logs:
    persistence:
      enabled: true
  # Airflow - DAGs Configs
  dags:
    # configs for the DAG git repository & sync container
    gitSync:
      enabled: true
      # url of the git repository
      repo: "git@github.com:mboglesby/airflow-dev.git"
      # the branch/tag/sha1 which we clone
      branch: master
      revision: HEAD
      # the name of a pre-created secret containing files for ~/.ssh/
      #
      # NOTE:
## - this is ONLY RELEVANT for SSH git repos
## - the secret commonly includes files: id_rsa, id_rsa.pub, known_hosts
## - known_hosts is NOT NEEDED if `git.sshKeyscan` is true

sshSecret: "airflow-ssh-git-secret"
## the name of the private key file in your `git.secret`
##
## NOTE:
## - this is ONLY RELEVANT for PRIVATE SSH git repos

sshSecretKey: id_rsa
## the git sync interval in seconds
##
syncWait: 60

EOF

$ helm install airflow airflow-stable/airflow -n airflow --version 8.0.8 --values ./custom-values.yaml
...

Congratulations. You have just deployed Apache Airflow!
1. Get the Airflow Service URL by running these commands:
   ```bash
   export NODE_PORT=$(kubectl get --namespace airflow -o jsonpath='{.spec.ports[0].nodePort}' services airflow-web)
   export NODE_IP=$(kubectl get nodes --namespace airflow -o jsonpath='{.items[0].status.addresses[0].address}')
   echo http://$NODE_IP:$NODE_PORT/
   ```
2. Open Airflow in your web browser

2. Confirm that all Airflow pods are up and running. It may take a few minutes for all pods to start.

```
$ kubectl -n airflow get pod
NAME                                READY   STATUS    RESTARTS   AGE
airflow-flower-b5656d44f-h8qjk      1/1     Running   0          2h
airflow-postgresql-0                1/1     Running   0          2h
airflow-redis-master-0              1/1     Running   0          2h
airflow-scheduler-9d95fcd9-clf4b     2/2     Running   2          2h
airflow-web-59c94db9c5-z7rg4         1/1     Running   0          2h
airflow-worker-0                    2/2     Running   2          2h
```

3. Obtain the Airflow web service URL by following the instructions that were printed to the console when you deployed Airflow using Helm in step 1.
$ export NODE_PORT=$(kubectl get --namespace airflow -o jsonpath='{.spec.ports[0].nodePort}' services airflow-web)
$ export NODE_IP=$(kubectl get nodes --namespace airflow -o jsonpath='{.items[0].status.addresses[0].address}')
$ echo http://$NODE_IP:$NODE_PORT/

4. Confirm that you can access the Airflow web service.

Example Apache Airflow Workflows

The NetApp Data Science Toolkit for Kubernetes can be used in conjunction with Airflow. Using the NetApp Data Science Toolkit with Airflow enables you to incorporate NetApp data management operations into automated workflows that are orchestrated by Airflow.

Refer to the Airflow Examples section within the NetApp Data Science Toolkit GitHub repository for details on using the toolkit with Airflow.
Example Trident Operations

This section includes examples of various operations that you may want to perform with Trident.

Import an Existing Volume

If there are existing volumes on your NetApp storage system/platform that you want to mount on containers within your Kubernetes cluster, but that are not tied to PVCs in the cluster, then you must import these volumes. You can use the Trident volume import functionality to import these volumes.

The example commands that follow show the importing of the same volume, named `pb_fg_all`, twice, once for each Trident Backend that was created in the example in the section Example Trident Backends for ONTAP AI Deployments, step 1. Importing the same volume twice in this manner enables you to mount the volume (an existing FlexGroup volume) multiple times across different LIFs, as described in the section Example Trident Backends for ONTAP AI Deployments, step 1. For more information about PVCs, see the official Kubernetes documentation. For more information about the volume import functionality, see the Trident documentation.

An `accessModes` value of `ReadOnlyMany` is specified in the example PVC spec files. For more information about the `accessMode` field, see the official Kubernetes documentation.

```
$ cat << EOF > ./pvc-import-pb_fg_all-iface1.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pb-fg-all-iface1
  namespace: default
spec:
  accessModes:
  - ReadOnlyMany
  storageClassName: ontap-ai-flexgroups-retain-iface1
EOF
```

```
$ tridentctl import volume ontap-ai-flexgroups-iface1 pb_fg_all -f ./pvc-import-pb_fg_all-iface1.yaml -n trident
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>SIZE</th>
<th>STORAGE CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

The Backend names that are specified in the following example import commands correspond to the Backends that were created in the example in the section Example Trident Backends for ONTAP AI Deployments, step 1. The StorageClass names that are specified in the following example PVC definition files correspond to the StorageClasses that were created in the example in the section Example Kubernetes StorageClasses for ONTAP AI Deployments, step 1.

```
$ tridentctl import volume ontap-ai-flexgroups-iface1 pb_fg_all -f ./pvc-import-pb_fg_all-iface1.yaml -n trident
```

```
$ cat << EOF > ./pvc-import-pb_fg_all-iface2.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pb-fg-all-iface2
  namespace: default
spec:
  accessModes:
  - ReadOnlyMany
  storageClassName: ontap-ai-flexgroups-retain-iface2
EOF
```

```
$ tridentctl import volume ontap-ai-flexgroups-iface2 pb_fg_all -f ./pvc-import-pb_fg_all-iface2.yaml -n trident
```

```
+--------------------------------+--------+---------+
+--------------------------------+--------+---------+
+---------------------------------+--------+---------+
|          NAME                  |  SIZE  |       STORAGE CLASS |
| PROTOCOL | BACKEND UUID | STATE |
+----------|--------------+--------|
+----------|--------------+--------|
+----------|--------------+--------|
$ cat << EOF > ./pvc-import-pb_fg_all-iface2.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pb-fg-all-iface2
  namespace: default
spec:
  accessModes:
  - ReadOnlyMany
  storageClassName: ontap-ai-flexgroups-retain-iface2
EOF
$ tridentctl import volume ontap-ai-flexgroups-iface2 pb_fg_all -f ./pvc-
import-pb_fg_all-iface2.yaml -n trident

$ tridentctl get volume -n trident

$ tridentctl get volume -n trident
Provision a New Volume

You can use Trident to provision a new volume on your NetApp storage system or platform. The following example commands show the provisioning of a new FlexVol volume. In this example, the volume is provisioned using the StorageClass that was created in the example in the section Example Kubernetes StorageClasses for ONTAP AI Deployments, step 2.

An accessModes value of ReadWriteMany is specified in the following example PVC definition file. For more information about the accessMode field, see the official Kubernetes documentation.
Example High-performance Jobs for ONTAP AI Deployments

This section includes examples of various high-performance jobs that can be executed when Kubernetes is deployed on an ONTAP AI pod.

Example High-performance Jobs for ONTAP AI Deployments

This section includes examples of various high-performance jobs that can be executed when Kubernetes is deployed on an ONTAP AI pod.

Execute a Single-Node AI Workload

To execute a single-node AI and ML job in your Kubernetes cluster, perform the following tasks from the deployment jump host. With Trident, you can quickly and easily make a data volume, potentially containing petabytes of data, accessible to a Kubernetes workload. To make such a data volume accessible from within a Kubernetes pod, simply specify a PVC in the pod definition. This step is a Kubernetes-native operation; no NetApp expertise is required.
This section assumes that you have already containerized (in the Docker container format) the specific AI and ML workload that you are attempting to execute in your Kubernetes cluster.

1. The following example commands show the creation of a Kubernetes job for a TensorFlow benchmark workload that uses the ImageNet dataset. For more information about the ImageNet dataset, see the ImageNet website.

This example job requests eight GPUs and therefore can run on a single GPU worker node that features eight or more GPUs. This example job could be submitted in a cluster for which a worker node featuring eight or more GPUs is not present or is currently occupied with another workload. If so, then the job remains in a pending state until such a worker node becomes available.

Additionally, in order to maximize storage bandwidth, the volume that contains the needed training data is mounted twice within the pod that this job creates. Another volume is also mounted in the pod. This second volume will be used to store results and metrics. These volumes are referenced in the job definition by using the names of the PVCs. For more information about Kubernetes jobs, see the official Kubernetes documentation.

An emptyDir volume with a medium value of Memory is mounted to /dev/shm in the pod that this example job creates. The default size of the /dev/shm virtual volume that is automatically created by the Docker container runtime can sometimes be insufficient for TensorFlow's needs. Mounting an emptyDir volume as in the following example provides a sufficiently large /dev/shm virtual volume. For more information about emptyDir volumes, see the official Kubernetes documentation.

The single container that is specified in this example job definition is given a securityContext > privileged value of true. This value means that the container effectively has root access on the host. This annotation is used in this case because the specific workload that is being executed requires root access. Specifically, a clear cache operation that the workload performs requires root access. Whether or not this privileged: true annotation is necessary depends on the requirements of the specific workload that you are executing.

```
$ cat << EOF > ./netapp-tensorflow-single-imagenet.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-tensorflow-single-imagenet
spec:
  backoffLimit: 5
  template:
    spec:
      volumes:
      - name: dshm
        emptyDir:
          medium: Memory
      - name: testdata-iface1
        persistentVolumeClaim:
          claimName: pb-fg-all-iface1
      - name: testdata-iface2
        persistentVolumeClaim:
          claimName: pb-fg-all-iface2
EOF
```
2. Confirm that the job that you created in step 1 is running correctly. The following example command confirms that a single pod was created for the job, as specified in the job definition, and that this pod is currently running on one of the GPU worker nodes.

```
$ kubectl get pods -o wide
NAME                           READY STATUS
IP                             NODE       NOMINATED NODE
netapp-tensorflow-single-imagenet-m7x92   1/1   Running  0
3m  10.233.68.61  10.61.218.154  <none>
```

3. Confirm that the job that you created in step 1 completes successfully. The following example commands confirm that the job completed successfully.
$ kubectl get jobs
NAME                             COMPLETIONS   DURATION
AGE
netapp-tensorflow-single-imagenet 1/1           5m42s
10m

$ kubectl get pods
NAME                                                   READY   STATUS
RESTARTS   AGE
netapp-tensorflow-single-imagenet-m7x92                0/1     Completed
0          11m

$ kubectl logs netapp-tensorflow-single-imagenet-m7x92
[netapp-tensorflow-single-imagenet-m7x92:00008] PMIX ERROR: NO
PERMISSIONS in file gds_dstore.c at line 702
[netapp-tensorflow-single-imagenet-m7x92:00008] PMIX ERROR: NO
PERMISSIONS in file gds_dstore.c at line 711
Total images/sec = 6530.59125
================ Clean Cache !!! ==================
mpirun -allow-run-as-root -np 1 -H localhost:1 bash -c 'sync; echo 1 >
/proc/sys/vm/drop_caches'
=========================================
mpirun -allow-run-as-root -np 8 -H localhost:8 -bind-to none -map-by
slot -x NCCL_DEBUG=INFO -x LD_LIBRARY_PATH -x PATH python
/netapp/tensorflow/benchmarks_190205/scripts/tf_cnn_benchmarks/tf_cnn_be
chnmarks.py --model=resnet50 --batch_size=256 --device=gpu
--force_gpu_compatible=True --num_intra_threads=1 --num_inter_threads=48
--variable_update=horovod --batch_group_size=20 --num_batches=500
--nondistortions --num_gpus=1 --data_format=NCHW --use_fp16=True
--use_tf_layers=False --data_name=imagenet --use_datasets=True
--data_dir=/mnt/mount_0/dataset/imagenet
--datasets_parallel_interleave_cycle_length=10
--datasets_sloppy_parallel_interleave=False --num_mounts=2
--mount_prefix=/mnt/mount_%d --datasets_prefetch_buffer_size=2000
--datasets_use_prefetch=True --datasets_num_private_threads=4
--horovod_device=gpu >
/tmp/20190814_105450_tensorflow_horovod_rdma_resnet50_gpu_8_256_b500_ima
genet_nodistort_fp16_r10_m2_nockpt.txt 2>&1

4. **Optional:** Clean up job artifacts. The following example commands show the deletion of the job object that
was created in step 1.

When you delete the job object, Kubernetes automatically deletes any associated pods.
Execute a Synchronous Distributed AI Workload

To execute a synchronous multinode AI and ML job in your Kubernetes cluster, perform the following tasks on the deployment jump host. This process enables you to take advantage of data that is stored on a NetApp volume and to use more GPUs than a single worker node can provide. See the following figure for a depiction of a synchronous distributed AI job.

Synchronous distributed jobs can help increase performance and training accuracy compared with asynchronous distributed jobs. A discussion of the pros and cons of synchronous jobs versus asynchronous jobs is outside the scope of this document.

1. The following example commands show the creation of one worker that participates in the synchronous distributed execution of the same TensorFlow benchmark job that was executed on a single node in the example in the section Execute a Single-Node AI Workload. In this specific example, only a single worker is deployed because the job is executed across two worker nodes.
This example worker deployment requests eight GPUs and thus can run on a single GPU worker node that features eight or more GPUs. If your GPU worker nodes feature more than eight GPUs, to maximize performance, you might want to increase this number to be equal to the number of GPUs that your worker nodes feature. For more information about Kubernetes deployments, see the official Kubernetes documentation.

A Kubernetes deployment is created in this example because this specific containerized worker would never complete on its own. Therefore, it doesn’t make sense to deploy it by using the Kubernetes job construct. If your worker is designed or written to complete on its own, then it might make sense to use the job construct to deploy your worker.

The pod that is specified in this example deployment specification is given a hostNetwork value of true. This value means that the pod uses the host worker node’s networking stack instead of the virtual networking stack that Kubernetes usually creates for each pod. This annotation is used in this case because the specific workload relies on Open MPI, NCCL, and Horovod to execute the workload in a synchronous distributed manner. Therefore, it requires access to the host networking stack. A discussion about Open MPI, NCCL, and Horovod is outside the scope of this document. Whether or not this hostNetwork: true annotation is necessary depends on the requirements of the specific workload that you are executing. For more information about the hostNetwork field, see the official Kubernetes documentation.

```yaml
$ cat << EOF > ./netapp-tensorflow-multi-imagenet-worker.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: netapp-tensorflow-multi-imagenet-worker
spec:
  replicas: 1
  selector:
    matchLabels:
      app: netapp-tensorflow-multi-imagenet-worker
  template:
    metadata:
      labels:
        app: netapp-tensorflow-multi-imagenet-worker
    spec:
      hostNetwork: true
      volumes:
      - name: dshm
        emptyDir:
          medium: Memory
      - name: testdata-iface1
        persistentVolumeClaim:
          claimName: pb-fg-all-iface1
      - name: testdata-iface2
        persistentVolumeClaim:
          claimName: pb-fg-all-iface2
      - name: results
        persistentVolumeClaim:
```
2. Confirm that the worker deployment that you created in step 1 launched successfully. The following example commands confirm that a single worker pod was created for the deployment, as indicated in the deployment definition, and that this pod is currently running on one of the GPU worker nodes.

```
$ kubectl get pods -o wide
NAME                      READY
STATUS         RESTARTS AGE
IP               NODE            NOMINATED NODE
netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725 1/1
Running 0 60s 10.61.218.154 10.61.218.154 <none>
$ kubectl logs netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725
22122
```

3. Create a Kubernetes job for a master that kicks off, participates in, and tracks the execution of the synchronous multinode job. The following example commands create one master that kicks off, participates in, and tracks the synchronous distributed execution of the same TensorFlow benchmark job that was executed on a single node in the example in the section Execute a Single-Node AI Workload.
This example master job requests eight GPUs and thus can run on a single GPU worker node that features eight or more GPUs. If your GPU worker nodes feature more than eight GPUs, to maximize performance, you might want to increase this number to be equal to the number of GPUs that your worker nodes feature.

The master pod that is specified in this example job definition is given a `hostNetwork` value of `true`, just as the worker pod was given a `hostNetwork` value of `true` in step 1. See step 1 for details about why this value is necessary.

```bash
$ cat << EOF > ./netapp-tensorflow-multi-imagenet-master.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-tensorflow-multi-imagenet-master
spec:
  backoffLimit: 5
  template:
    spec:
      hostNetwork: true
      volumes:
      - name: dshm
        emptyDir:
          medium: Memory
      - name: testdata-iface1
        persistentVolumeClaim:
          claimName: pb-fg-all-iface1
      - name: testdata-iface2
        persistentVolumeClaim:
          claimName: pb-fg-all-iface2
      - name: results
        persistentVolumeClaim:
          claimName: tensorflow-results
      containers:
      - name: netapp-tensorflow-py2
        image: netapp/tensorflow-py2:19.03.0
        resources:
          limits:
            nvidia.com/gpu: 8
        volumeMounts:
        - mountPath: /dev/shm
          name: dshm
        - mountPath: /mnt/mount_0
          name: testdata-iface1
        - mountPath: /mnt/mount_1
```

124
4. Confirm that the master job that you created in step 3 is running correctly. The following example command confirms that a single master pod was created for the job, as indicated in the job definition, and that this pod is currently running on one of the GPU worker nodes. You should also see that the worker pod that you originally saw in step 1 is still running and that the master and worker pods are running on different nodes.

```
$ kubectl get pods -o wide
```

```
NAME                     STATUS     RESTARTS   AGE
IP: 10.61.218.152        1/1         45s
                  NODE: 10.61.218.152   NOMINATED NODE: <none>
netapp-tensorflow-multi-imagenet-master-ppwwj
Running 0              10.61.218.152
```

5. Confirm that the master job that you created in step 3 completes successfully. The following example commands confirm that the job completed successfully.

```
$ kubectl get jobs
```

```
NAME                                      COMPLETIONS   DURATION   AGE
netapp-tensorflow-multi-imagenet-master   1/1           5m50s      9m18s
```

```
$ kubectl logs netapp-tensorflow-multi-imagenet-master-ppwwj
```

```
rm: cannot remove '/lib': Is a directory
[10.61.218.154:00033] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at
```
[10.61.218.154:00033] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at line 711
[10.61.218.152:00008] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at line 702
[10.61.218.152:00008] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at line 711
Total images/sec = 12881.33875
============= Clean Cache !!! ==============
mpirun -allow-run-as-root -np 2 -H 10.61.218.152:1,10.61.218.154:1 -mca pml ob1 -mca btl ^openib -mca btl_tcp_if_include enp1s0f0 -mca plm_rsh_agent ssh -mca plm_rsh_args "-p 22122" bash -c 'sync; echo 1 > /proc/sys/vm/drop_caches'
================= Clean Cache !!! ==================
mpirun -allow-run-as-root -np 16 -H 10.61.218.152:8,10.61.218.154:8 -bind-to none -map-by slot -x NCCL_DEBUG=INFO -x LD_LIBRARY_PATH -x PATH -mca pml ob1 -mca btl ^openib -mca btl_tcp_if_include enp1s0f0 -x NCCL_IB_HCA=mlx5 -x NCCL_NET_GDR_READ=1 -x NCCL_IB_SL=3 -x NCCL_IB_GID_INDEX=3 -x NCCL_SOCKET_IFNAME=enp5s0.3091,enp12s0.3092,enp132s0.3093,enp139s0.3094 -x NCCL_IB_CUDA_SUPPORT=1 -mca orte_base_help_aggregate 0 -mca plm_rsh_agent ssh -mca plm_rsh_args "-p 22122" python /netapp/tensorflow/benchmarks_190205/scripts/tf_cnn_benchmarks/tf_cnn_benchmarks.py --model=resnet50 --batch_size=256 --device=gpu --force_gpu_compatible=True --num_intra_threads=1 --num_inter_threads=48 --variable_update=horovod --batch_group_size=20 --num_batches=500 --nodistortions --num_gpus=1 --data_format=NCHW --use_fp16=True --use_tf_layers=False --data_name=imagenet --use_datasets=True --data_dir=/mnt/mount_0/dataset/imagenet --datasets_parallel_interleave_cycle_length=10 --datasets_sloppy_parallel_interleave=False --num_mounts=2 --mount_prefix=/mnt/mount_%d --datasets_prefetch_buffer_size=2000 --datasets_use_prefetch=True --datasets_num_private_threads=4 --horovod_device=gpu > /tmp/20190814_161609_tensorflow_horovod_rdma_resnet50_gpu_16_256_b500_imagenet_nodistort_fp16_r10_m2_nockpt.txt 2>&1

6. Delete the worker deployment when you no longer need it. The following example commands show the deletion of the worker deployment object that was created in step 1.

When you delete the worker deployment object, Kubernetes automatically deletes any associated worker pods.
$ kubectl get deployments
NAME                                    DESIRED CURRENT AVAILABLE AGE
netapp-tensorflow-multi-imagenet-worker 1      1      1 1 43m

$ kubectl get pods
NAME                                    READY
STATUS       RESTARTS   AGE
netapp-tensorflow-multi-imagenet-master-ppwwj 0/1
Completed 0 17m
netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725 1/1
Running 0 43m

$ kubectl delete deployment netapp-tensorflow-multi-imagenet-worker
deployment.extensions "netapp-tensorflow-multi-imagenet-worker" deleted

$ kubectl get deployments
No resources found.

$ kubectl get pods
NAME                                    READY STATUS
RESTARTS   AGE
netapp-tensorflow-multi-imagenet-master-ppwwj 0/1 Completed 0
18m

7. **Optional**: Clean up the master job artifacts. The following example commands show the deletion of the master job object that was created in step 3.

When you delete the master job object, Kubernetes automatically deletes any associated master pods.

$ kubectl get jobs
NAME                                    COMPLETIONS DURATION   AGE
netapp-tensorflow-multi-imagenet-master 1/1 5m50s 19m

$ kubectl get pods
NAME                                    READY STATUS
RESTARTS   AGE
netapp-tensorflow-multi-imagenet-master-ppwwj 0/1 Completed 0
19m

$ kubectl delete job netapp-tensorflow-multi-imagenet-master
job.batch "netapp-tensorflow-multi-imagenet-master" deleted

$ kubectl get jobs
No resources found.

$ kubectl get pods
No resources found.
Performance Testing

We performed a simple performance comparison as part of the creation of this solution. We executed several standard NetApp AI benchmarking jobs by using Kubernetes, and we compared the benchmark results with executions that were performed by using a simple Docker run command. We did not see any noticeable differences in performance. Therefore, we concluded that the use of Kubernetes to orchestrate containerized AI training jobs does not adversely affect performance. See the following table for the results of our performance comparison.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Dataset</th>
<th>Docker Run (images/sec)</th>
<th>Kubernetes (images/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-node TensorFlow</td>
<td>Synthetic data</td>
<td>6,667.2475</td>
<td>6,661.93125</td>
</tr>
<tr>
<td>Single-node TensorFlow</td>
<td>ImageNet</td>
<td>6,570.2025</td>
<td>6,530.59125</td>
</tr>
<tr>
<td>Synchronous distributed two-node TensorFlow</td>
<td>Synthetic data</td>
<td>13,213.70625</td>
<td>13,218.288125</td>
</tr>
<tr>
<td>Synchronous distributed two-node TensorFlow</td>
<td>ImageNet</td>
<td>12,941.69125</td>
<td>12,881.33875</td>
</tr>
</tbody>
</table>

Conclusion

Companies and organizations of all sizes and across all industries are turning to artificial intelligence (AI), machine learning (ML), and deep learning (DL) to solve real-world problems, deliver innovative products and services, and to get an edge in an increasingly competitive marketplace. As organizations increase their use of AI, ML, and DL, they face many challenges, including workload scalability and data availability. These challenges can be addressed through the use of the NetApp AI Control Plane solution.

This solution enables you to rapidly clone a data namespace. Additionally, it allows you to define and implement AI, ML, and DL training workflows that incorporate the near-instant creation of data and model baselines for traceability and versioning. With this solution, you can trace every single model training run back to the exact dataset(s) that the model was trained and/or validated with. Lastly, this solution enables you to swiftly provision Jupyter Notebook workspaces with access to massive datasets.

Because this solution is targeted towards data scientists and data engineers, minimal NetApp or NetApp ONTAP expertise is required. With this solution, data management functions can be executed using simple and familiar tools and interfaces. Furthermore, this solution utilizes fully open-source and free components. Therefore, if you already have NetApp storage in your environment, you can implement this solution today. If you want to test drive this solution but you do not have already have NetApp storage, visit cloud.netapp.com, and you can be up and running with a cloud-based NetApp storage solution in no time.

MLRun Pipeline with Iguazio

TR-4834: NetApp and Iguazio for MLRun Pipeline

Rick Huang, David Arnette, NetApp
Marcelo Litovsky, Iguazio
This document covers the details of the MLRun pipeline using NetApp ONTAP AI, NetApp AI Control Plane, NetApp Cloud Volumes software, and the Iguazio Data Science Platform. We used Nuclio serverless function, Kubernetes Persistent Volumes, NetApp Cloud Volumes, NetApp Snapshot copies, Grafana dashboard, and other services on the Iguazio platform to build an end-to-end data pipeline for the simulation of network failure detection. We integrated Iguazio and NetApp technologies to enable fast model deployment, data replication, and production monitoring capabilities on premises as well as in the cloud.

The work of a data scientist should be focused on the training and tuning of machine learning (ML) and artificial intelligence (AI) models. However, according to research by Google, data scientists spend ~80% of their time figuring out how to make their models work with enterprise applications and run at scale, as shown in the following image depicting model development in the AI/ML workflow.

To manage end-to-end AI/ML projects, a wider understanding of enterprise components is needed. Although DevOps have taken over the definition, integration, and deployment these types of components, machine learning operations target a similar flow that includes AI/ML projects. To get an idea of what an end-to-end AI/ML pipeline touches in the enterprise, see the following list of required components:

- Storage
- Networking
- Databases
- File systems
- Containers
- Continuous integration and continuous deployment (CI/CD) pipeline
- Development integrated development environment (IDE)
In this paper, we demonstrate how the partnership between NetApp and Iguazio drastically simplifies the development of an end-to-end AI/ML pipeline. This simplification accelerates the time to market for all of your AI/ML applications.

Target Audience

The world of data science touches multiple disciplines in information technology and business.

- The data scientist needs the flexibility to use their tools and libraries of choice.
- The data engineer needs to know how the data flows and where it resides.
- A DevOps engineer needs the tools to integrate new AI/ML applications into their CI/CD pipelines.
- Business users want to have access to AI/ML applications. We describe how NetApp and Iguazio help each of these roles bring value to business with our platforms.

Solution Overview

This solution follows the lifecycle of an AI/ML application. We start with the work of data scientists to define the different steps needed to prep data and train and deploy models. We follow with the work needed to create a full pipeline with the ability to track artifacts, experiment with execution, and deploy to Kubeflow. To complete the full cycle, we integrate the pipeline with NetApp Cloud Volumes to enable data versioning, as seen in the following image.
Technology Overview

NetApp Overview

NetApp is the data authority for the hybrid cloud. NetApp provides a full range of hybrid cloud data services that simplify management of applications and data across cloud and on-premises environments to accelerate digital transformation. Together with our partners, NetApp empowers global organizations to unleash the full potential of their data to expand customer touch points, foster greater innovation, and optimize their operations.

NetApp ONTAP AI

NetApp ONTAP AI, powered by NVIDIA DGX systems and NetApp cloud-connected all-flash storage, streamlines the flow of data reliably and speeds up analytics, training, and inference with your data fabric that spans from edge to core to cloud. It gives IT organizations an architecture that provides the following benefits:

• Eliminates design complexities
• Allows independent scaling of compute and storage
• Enables customers to start small and scale seamlessly
• Offers a range of storage options for various performance and cost points

NetApp ONTAP AI offers converged infrastructure stacks incorporating NVIDIA DGX-1, a petaflop-scale AI system, and NVIDIA Mellanox high-performance Ethernet switches to unify AI workloads, simplify deployment, and accelerate ROI. We leveraged ONTAP AI with one DGX-1 and NetApp AFF A800 storage system for this technical report. The following image shows the topology of ONTAP AI with the DGX-1 system used in this validation.
NetApp AI Control Plane

The NetApp AI Control Plane enables you to unleash AI and ML with a solution that offers extreme scalability, streamlined deployment, and nonstop data availability. The AI Control Plane solution integrates Kubernetes and Kubeflow with a data fabric enabled by NetApp. Kubernetes, the industry-standard container orchestration platform for cloud-native deployments, enables workload scalability and portability. Kubeflow is an open-source machine-learning platform that simplifies management and deployment, enabling developers to do more data science in less time. A data fabric enabled by NetApp offers uncompromising data availability and portability to make sure that your data is accessible across the pipeline, from edge to core to cloud. This technical report uses the NetApp AI Control Plane in an MLRun pipeline. The following image shows Kubernetes cluster management page where you can have different endpoints for each cluster. We connected NFS Persistent Volumes to the Kubernetes cluster, and the following images show an Persistent Volume connected to the cluster, where NetApp Trident offers persistent storage support and data management capabilities.
Persistent Volumes for Kubernetes

Connected with Kubernetes Cluster
Cloud Volumes ONTAP is connected to 1 Kubernetes cluster. View Cluster

You can connect another Kubernetes cluster to this Cloud Volumes ONTAP system. If the Kubernetes cluster is in a different network than Cloud Volumes ONTAP, specify a custom export policy to provide access to clients.

Kubernetes Cluster | Custom Export Policy (Optional)

Select Kubernetes Cluster
kubernetes

Custom Export Policy
172.31.0.0/16

Set as default storage class
- NFS  - ISCSI

Connect  Cancel
Iguazio Overview

The Iguazio Data Science Platform is a fully integrated and secure data science platform as a service (PaaS) that simplifies development, accelerates performance, facilitates collaboration, and addresses operational challenges. This platform incorporates the following components, and the Iguazio Data Science Platform is presented in the following image:

- A data-science workbench that includes Jupyter Notebooks, integrated analytics engines, and Python packages
- Model management with experiments tracking and automated pipeline capabilities
- Managed data and ML services over a scalable Kubernetes cluster
- Nuclio, a real-time serverless functions framework
- An extremely fast and secure data layer that supports SQL, NoSQL, time-series databases, files (simple objects), and streaming
- Integration with third-party data sources such as NetApp, Amazon S3, HDFS, SQL databases, and streaming or messaging protocols
- Real-time dashboards based on Grafana
Software and Hardware Requirements

Network Configuration

The following is the network configuration requirement for setting up in the cloud:

- The Iguazio cluster and NetApp Cloud Volumes must be in the same virtual private cloud.
- The cloud manager must have access to port 6443 on the Iguazio app nodes.
- We used Amazon Web Services in this technical report. However, users have the option of deploying the solution in any Cloud provider. For on-premises testing in ONTAP AI with NVIDIA DGX-1, we used the Iguazio hosted DNS service for convenience.

Clients must be able to access dynamically created DNS domains. Customers can use their own DNS if desired.

Hardware Requirements

You can install Iguazio on-premises in your own cluster. We have verified the solution in NetApp ONTAP AI with an NVIDIA DGX-1 system. The following table lists the hardware used to test this solution:

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGX-1 systems</td>
<td>1</td>
</tr>
<tr>
<td>NetApp AFF A800 system</td>
<td>1 high-availability (HA) pair, includes 2 controllers and 48 NVMe SSDs (3.8TB or above)</td>
</tr>
<tr>
<td>Cisco Nexus 3232C network switches</td>
<td>2</td>
</tr>
</tbody>
</table>

The following table lists the software components required for on-premise testing:
This solution was fully tested with Iguazio version 2.5 and NetApp Cloud Volumes ONTAP for AWS. The Iguazio cluster and NetApp software are both running on AWS.

### Software

<table>
<thead>
<tr>
<th>Software</th>
<th>Version or Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iguazio</td>
<td>Version 2.8+</td>
</tr>
<tr>
<td>App node</td>
<td>M5.4xlarge</td>
</tr>
<tr>
<td>Data node</td>
<td>I3.4xlarge</td>
</tr>
</tbody>
</table>

### Network Device Failure Prediction Use Case Summary

This use case is based on an Iguazio customer in the telecommunications space in Asia. With 100K enterprise customers and 125k network outage events per year, there was a critical need to predict and take proactive action to prevent network failures from affecting customers. This solution provided them with the following benefits:

- Predictive analytics for network failures
- Integration with a ticketing system
- Taking proactive action to prevent network failures

As a result of this implementation of Iguazio, 60% of failures were proactively prevented.

### Setup Overview

Iguazio can be installed on-premises or on a cloud provider.

#### Iguazio Installation

Provisioning can be done as a service and managed by Iguazio or by the customer. In both cases, Iguazio provides a deployment application (Provazio) to deploy and manage clusters.

For on-premises installation, please refer to NVA-1121 for compute, network, and storage setup. On-premises deployment of Iguazio is provided by Iguazio without additional cost to the customer. See this page for DNS and SMTP server configurations. The Provazio installation page is shown as follows.

---

This page contains information about various software versions and configurations, including:

- NetApp ONTAP data management software version 9.7
- Cisco NX-OS switch firmware version 7.0(3)I6(1)
- NVIDIA DGX OS version 4.4 - Ubuntu 18.04 LTS
- Docker container platform version 19.03.5
- Container version 20.01-tf1-py2
- Machine learning framework TensorFlow 1.15.0
- ESX Server version 6.5

---

This solution was fully tested with Iguazio version 2.5 and NetApp Cloud Volumes ONTAP for AWS. The Iguazio cluster and NetApp software are both running on AWS.
Configuring Kubernetes Cluster

This section is divided into two parts for cloud and on-premises deployment respectively.

Cloud Deployment Kubernetes Configuration

Through NetApp Cloud Manager, you can define the connection to the Iguazio Kubernetes cluster. Trident requires access to multiple resources in the cluster to make the volume available.

1. To enable access, obtain the Kubernetes config file from one the Iguazio nodes. The file is located under `/home/Iguazio/.kube/config`. Download this file to your desktop.
2. Go to Discover Cluster to configure.
3. Upload the Kubernetes config file. See the following image.

Upload Kubernetes Configuration File

Upload the Kubernetes configuration file (kubeconfig) so Cloud Manager can install Trident on the Kubernetes cluster.

Connecting Cloud Volumes ONTAP with a Kubernetes cluster enables users to request and manage persistent volumes using native Kubernetes interfaces and constructs. Users can take advantage of ONTAP’s advanced data management features without having to know anything about it. Storage provisioning is enabled by using NetApp Trident. Learn more about Trident for Kubernetes.

Upload File

4. Deploy Trident and associate a volume with the cluster. See the following image on defining and assigning a Persistent Volume to the Iguazio cluster. This process creates a Persistent Volume (PV) in Iguazio’s Kubernetes cluster. Before you can use it, you must define a Persistent Volume Claim (PVC).
Persistent Volumes for Kubernetes

**Connected with Kubernetes Cluster**

Cloud Volumes ONTAP is connected to 1 Kubernetes cluster. View Cluster

You can connect another Kubernetes cluster to this Cloud Volumes ONTAP system. If the Kubernetes cluster is in a different network than Cloud Volumes ONTAP, specify a custom export policy to provide access to clients.

**Kubernetes Cluster**

**Custom Export Policy (Optional)**

- **Select Kubernetes Cluster**
  - kubernetes

- **Custom Export Policy**
  - 172.31.0.0/16

- **Set as default storage class**
  - ○ NFS  ○ iSCSI

**On-Premises Deployment Kubernetes Configuration**

For on-premises installation of NetApp Trident, see TR-4798 for details. After configuring your Kubernetes cluster and installing NetApp Trident, you can connect Trident to the Iguazio cluster to enable NetApp data management capabilities, such as taking Snapshot copies of your data and model.

**Define Persistent Volume Claim**

1. Save the following YAML to a file to create a PVC of type Basic.

   ```yaml
   kind: PersistentVolumeClaim
   apiVersion: v1
   metadata:
     name: basic
   spec:
     accessModes:
     - ReadWriteOnce
     resources:
       requests:
       storage: 100Gi
     storageClassName: netapp-file
   ```

2. Apply the YAML file to your Iguazio Kubernetes cluster.
Kubectl -n default-tenant apply -f <your yaml file>

Attach NetApp Volume to the Jupyter Notebook

Iguazio offers several managed services to provide data scientists with a full end-to-end stack for development and deployment of AI/ML applications. You can read more about these components at the Iguazio Overview of Application Services and Tools.

One of the managed services is Jupyter Notebook. Each developer gets its own deployment of a notebook container with the resources they need for development. To give them access to the NetApp Cloud Volume, you can assign the volume to their container and resource allocation, running user, and environment variable settings for Persistent Volume Claims is presented in the following image.

For an on-premises configuration, you can refer to TR-4798 on the Trident setup to enable NetApp ONTAP data management capabilities, such as taking Snapshot copies of your data or model for versioning control. Add the following line in your Trident back-end config file to make Snapshot directories visible:

```json
{
  ...  
  "defaults": {
    "snapshotDir": "true"
  }
}
```

You must create a Trident back-end config file in JSON format, and then run the following Trident command to reference it:

```
tridentctl create backend -f <backend-file>
```

Deploying the Application

The following sections describe how to install and deploy the application.
Get Code from GitHub

Now that the NetApp Cloud Volume or NetApp Trident volume is available to the Iguazio cluster and the developer environment, you can start reviewing the application.

Users have their own workspace (directory). On every notebook, the path to the user directory is /User. The Iguazio platform manages the directory. If you follow the instructions above, the NetApp Cloud volume is available in the /netapp directory.

Get the code from GitHub using a Jupyter terminal.

At the Jupyter terminal prompt, clone the project.

```
  cd /User
  git clone .
```

You should now see the netops-netapp folder on the file tree in Jupyter workspace.

Configure Working Environment

Copy the Notebook set_env-Example.ipynb as set_env.ipynb. Open and edit set_env.ipynb. This notebook sets variables for credentials, file locations, and execution drivers.

If you follow the instructions above, the following steps are the only changes to make:

1. Obtain this value from the Iguazio services dashboard: docker_registry

   Example: docker-registry.default-tenant.app.clusterc.clusterq.iguaziodev.com:80
2. Change admin to your Iguazio username:

    IGZ_CONTAINER_PATH = '/users/admin'

The following are the ONTAP system connection details. Include the volume name that was generated when Trident was installed. The following setting is for an on-premises ONTAP cluster:

    ontapClusterMgmtHostname = '0.0.0.0'
    ontapClusterAdminUsername = 'USER'
    ontapClusterAdminPassword = 'PASSWORD'
    sourceVolumeName = 'SOURCE VOLUME'

The following setting is for Cloud Volumes ONTAP:

    MANAGER=ontapClusterMgmtHostname
    svm='svm'
    email='email'
    password=ontapClusterAdminPassword
    weid = "weid"
    volume=sourceVolumeName

Create Base Docker Images

Everything you need to build an ML pipeline is included in the Iguazio platform. The developer can define the specifications of the Docker images required to run the pipeline and execute the image creation from Jupyter Notebook. Open the notebook `create_images.ipynb` and Run All Cells.

This notebook creates two images that we use in the pipeline.

* iguazio/netapp. Used to handle ML tasks.

Create image for training pipeline

```
[4]: fn.build_config(image='docker_registry+/iguazio/netapp', commands=['pip install \
v3io_frames ssspec==0.3.3 PyYAML==5.1.2 pyarrow==0.15.1 pandas==0.25.3 matplotlib seaborn yellowbr
fn.deploy()]
```

* netapp/pipeline. Contains utilities to handle NetApp Snapshot copies.

Create image for Ontap utilities

```
[9]: fn.build_config(image='docker_registry+/netapp/pipeline/latest', commands=['apt update', 'pip install v3io_frames netapp_ontap
fn.deploy()]
```

Review Individual Jupyter Notebooks

The following table lists the libraries and frameworks we used to build this task. All these components have been fully integrated with Iguazio’s role-based access and security controls.
<table>
<thead>
<tr>
<th>Libraries/Framework</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLRun</td>
<td>An managed by Iguazio to enable the assembly, execution, and monitoring of an ML/AI pipeline.</td>
</tr>
<tr>
<td>Nuclio</td>
<td>A serverless functions framework integrated with Iguazio. Also available as an open-source project managed by Iguazio.</td>
</tr>
<tr>
<td>Kubeflow</td>
<td>A Kubernetes-based framework to deploy the pipeline. This is also an open-source project to which Iguazio contributes. It is integrated with Iguazio for added security and integration with the rest of the infrastructure.</td>
</tr>
<tr>
<td>Docker</td>
<td>A Docker registry run as a service in the Iguazio platform. You can also change this to connect to your registry.</td>
</tr>
<tr>
<td>NetApp Cloud Volumes</td>
<td>Cloud Volumes running on AWS give us access to large amounts of data and the ability to take Snapshot copies to version the datasets used for training.</td>
</tr>
<tr>
<td>Trident</td>
<td>Trident is an open-source project managed by NetApp. It facilitates the integration with storage and compute resources in Kubernetes.</td>
</tr>
</tbody>
</table>

We used several notebooks to construct the ML pipeline. Each notebook can be tested individually before being brought together in the pipeline. We cover each notebook individually following the deployment flow of this demonstration application.

The desired result is a pipeline that trains a model based on a Snapshot copy of the data and deploys the model for inference. A block diagram of a completed MLRun pipeline is shown in the following image.
Deploy Data Generation Function

This section describes how we used Nuclio serverless functions to generate network device data. The use case is adapted from an Iguazio client that deployed the pipeline and used Iguazio services to monitor and predict network device failures.

We simulated data coming from network devices. Executing the Jupyter notebook `data-generator.ipynb` creates a serverless function that runs every 10 minutes and generates a Parquet file with new data. To deploy the function, run all the cells in this notebook. See the Nuclio website to review any unfamiliar components in this notebook.

A cell with the following comment is ignored when generating the function. Every cell in the notebook is assumed to be part of the function. Import the Nuclio module to enable `%nuclio magic`.

```python
# nuclio: ignore
import nuclio
```

In the spec for the function, we defined the environment in which the function executes, how it is triggered, and the resources it consumes.
spec = nuclio.ConfigSpec(config={"spec.triggers.inference.kind":"cron",
"spec.triggers.inference.attributes.interval":"10m",
    "spec.readinessTimeoutSeconds" : 60,
    "spec.minReplicas" : 1},......

The `init_context` function is invoked by the Nuclio framework upon initialization of the function.

```python
def init_context(context):
    ...
```

Any code not in a function is invoked when the function initializes. When you invoke it, a handler function is executed. You can change the name of the handler and specify it in the function spec.

```python
def handler(context, event):
    ...
```

You can test the function from the notebook prior to deployment.

```python
%%time
# nuclio: ignore
init_context(context)
event = nuclio.Event(body='')
output = handler(context, event)
output
```

The function can be deployed from the notebook or it can be deployed from a CI/CD pipeline (adapting this code).

```python
addr = nuclio.deploy_file(name='generator',project='netops',spec=spec, tag='v1.1')
```

**Pipeline Notebooks**

These notebooks are not meant to be executed individually for this setup. This is just a review of each notebook. We invoked them as part of the pipeline. To execute them individually, review the MLRun documentation to execute them as Kubernetes jobs.

**snap_cv.ipynb**

This notebook handles the Cloud Volume Snapshot copies at the beginning of the pipeline. It passes the name of the volume to the pipeline context. This notebook invokes a shell script to handle the Snapshot copy. While running in the pipeline, the execution context contains variables to help locate all files needed for execution.
While writing this code, the developer does not have to worry about the file location in the container that executes it. As described later, this application is deployed with all its dependencies, and it is the definition of the pipeline parameters that provides the execution context.

```python
command = os.path.join(context.get_param('APP_DIR'),"snap_cv.sh")
```

The created Snapshot copy location is placed in the MLRun context to be consumed by steps in the pipeline.

```python
context.log_result('snapVolumeDetails',snap_path)
```

The next three notebooks are run in parallel.

**data-prep.ipynb**

Raw metrics must be turned into features to enable model training. This notebook reads the raw metrics from the Snapshot directory and writes the features for model training to the NetApp volume.

When running in the context of the pipeline, the input `DATA_DIR` contains the Snapshot copy location.

```python
metrics_table = os.path.join(str(mlruncontext.get_input('DATA_DIR',
                          os.getenv('DATA_DIR','/netpp'))),
                          mlruncontext.get_param('metrics_table',
                          os.getenv('metrics_table','netops_metrics_parquet')))
```

**describe.ipynb**

To visualize the incoming metrics, we deploy a pipeline step that provides plots and graphs that are available through the Kubeflow and MLRun UIs. Each execution has its own version of this visualization tool.

```python
ax.set_title("features correlation")
plt.savefig(os.path.join(base_path, "plots/corr.png"))
context.log_artifact(PlotArtifact("correlation", body=plt.gcf()),
local_path="plots/corr.html")
```

**deploy-feature-function.ipynb**

We continuously monitor the metrics looking for anomalies. This notebook creates a serverless function that generates the features need to run prediction on incoming metrics. This notebook invokes the creation of the function. The function code is in the notebook `data-prep.ipynb`. Notice that we use the same notebook as a step in the pipeline for this purpose.

**training.ipynb**

After we create the features, we trigger the model training. The output of this step is the model to be used for inferencing. We also collect statistics to keep track of each execution (experiment).
For example, the following command enters the accuracy score into the context for that experiment. This value is visible in Kubeflow and MLRun.

```python
context.log_result('accuracy', score)
```

**deploy-inference-function.ipynb**

The last step in the pipeline is to deploy the model as a serverless function for continuous inferencing. This notebook invokes the creation of the serverless function defined in `nuclio-inference-function.ipynb`.

**Review and Build Pipeline**

The combination of running all the notebooks in a pipeline enables the continuous run of experiments to reassess the accuracy of the model against new metrics. First, open the `pipeline.ipynb` notebook. We take you through details that show how NetApp and Iguazio simplify the deployment of this ML pipeline.

We use MLRun to provide context and handle resource allocation to each step of the pipeline. The MLRun API service runs in the Iguazio platform and is the point of interaction with Kubernetes resources. Each developer cannot directly request resources; the API handles the requests and enables access controls.

```python
# MLRun API connection definition
mlconf.dbpath = 'http://mlrun-api:8080'
```

The pipeline can work with NetApp Cloud Volumes and on-premises volumes. We built this demonstration to use Cloud Volumes, but you can see in the code the option to run on-premises.
The first action needed to turn a Jupyter notebook into a Kubeflow step is to turn the code into a function. A function has all the specifications required to run that notebook. As you scroll down the notebook, you can see that we define a function for every step in the pipeline.

### Part of the Notebook

<table>
<thead>
<tr>
<th>Part of the Notebook</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;code_to_function&gt;</td>
<td>Name of the function: Name of the function: Project name. used to organize all project artifacts. Project name. used to organize all project artifacts. This is visible in the MLRun UI. This is visible in the MLRun UI. Kind. In this case, a Kubernetes job. This could be Dask, mpi, sparkk8s, and more. See the MLRun documentation for more details. Kind. In this case, a Kubernetes job. This could be Dask, mpi, sparkk8s, and more. See the MLRun documentation for more details. File. The name of the notebook. This can also be a location in Git (HTTP). File. The name of the notebook. This can also be a location in Git (HTTP).</td>
</tr>
<tr>
<td>(part of the MLRun module)</td>
<td></td>
</tr>
<tr>
<td>image</td>
<td>The name of the Docker image we are using for this step. We created this earlier with the create-image.ipynb notebook. The name of the Docker image we are using for this step. We created this earlier with the create-image.ipynb notebook.</td>
</tr>
<tr>
<td>volume_mounts &amp; volumes</td>
<td>Details to mount the NetApp Cloud Volume at run time. Details to mount the NetApp Cloud Volume at run time.</td>
</tr>
</tbody>
</table>

We also define parameters for the steps.
After you have the function definition for all steps, you can construct the pipeline. We use the `kfp` module to make this definition. The difference between using MLRun and building on your own is the simplification and shortening of the coding.

The functions we defined are turned into step components using the `as_step` function of MLRun.

### Snapshot Step Definition

Initiate a Snapshot function, output, and mount v3io as source:

```python
snap = snapfn.as_step(NewTask(handler='handler',params=snap_params),
    name='NetApp_Cloud_Volume_Snapshot',outputs=['snapVolumeDetails','training_parquet_file']).apply(mount_v3io())
```

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>NewTask</td>
<td>NewTask is the definition of the function run.</td>
</tr>
<tr>
<td>(MLRun module)</td>
<td>Handler. Name of the Python function to invoke. We used the name handler in the notebook, but it is not required.</td>
</tr>
<tr>
<td>params</td>
<td>params. The parameters we passed to the execution. Inside our code, we use <code>context.get_param('PARAMETER')</code> to get the values.</td>
</tr>
</tbody>
</table>
**Parameters** | **Details**
---|---
**as_step** | Name. Name of the Kubeflow pipeline step. 
**outputs**. These are the values that the step adds to the dictionary on completion. Take a look at the snap_cv.ipynb notebook. 
**mount_v3io()**. This configures the step to mount /User for the user executing the pipeline.

```python
prep = data_prep.as_step(name='data-prep',
                        handler='handler',
                        params=params,
                        inputs = {'DATA_DIR': snap.outputs['snapVolumeDetails']},
                        out_path=artifacts_path).apply(mount_v3io()).after(snap)
```

**Parameters** | **Details**
---|---
**inputs** | You can pass to a step the outputs of a previous step. In this case, snap.outputs['snapVolumeDetails'] is the name of the Snapshot copy we created on the snap step.
**out_path** | A location to place artifacts generating using the MLRun module log_artifacts.

You can run `pipeline.ipynb` from top to bottom. You can then go to the Pipelines tab from the Iguazio dashboard to monitor progress as seen in the Iguazio dashboard Pipelines tab.
Because we logged the accuracy of training step in every run, we have a record of accuracy for each experiment, as seen in the record of training accuracy.

<table>
<thead>
<tr>
<th>Run name</th>
<th>Status</th>
<th>Duration</th>
<th>Pipeline Version</th>
<th>Recurring</th>
<th>Start time</th>
<th>accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>xgb_pipeline 2020-03-24 18-51-08</td>
<td>✔️</td>
<td>0:08:43</td>
<td>[View pipeline]</td>
<td>-</td>
<td>3/24/2020, 2:51:09 PM</td>
<td>0.985</td>
</tr>
<tr>
<td>xgb_pipeline 2020-03-19 13-31-09</td>
<td>✔️</td>
<td>0:06:14</td>
<td>[View pipeline]</td>
<td>-</td>
<td>3/19/2020, 9:31:19 AM</td>
<td>0.980</td>
</tr>
<tr>
<td>xgb_pipeline 2020-03-18 12-56-09</td>
<td>✔️</td>
<td>0:08:11</td>
<td>[View pipeline]</td>
<td>-</td>
<td>3/18/2020, 8:56:08 AM</td>
<td>0.990</td>
</tr>
<tr>
<td>xgb_pipeline 2020-03-17 19-49-09</td>
<td>✔️</td>
<td>0:08:03</td>
<td>[View pipeline]</td>
<td>-</td>
<td>3/17/2020, 3:49:31 PM</td>
<td>0.985</td>
</tr>
<tr>
<td>xgb_pipeline 2020-03-17 18-34-08</td>
<td>✔️</td>
<td>0:05:54</td>
<td>[View pipeline]</td>
<td>-</td>
<td>3/17/2020, 2:34:56 PM</td>
<td>0.980</td>
</tr>
<tr>
<td>xgb_pipeline 2020-03-17 17-34-08</td>
<td>✔️</td>
<td>0:04:48</td>
<td>[View pipeline]</td>
<td>-</td>
<td>3/17/2020, 1:34:16 PM</td>
<td>0.992</td>
</tr>
<tr>
<td>xgb_pipeline 2020-03-17 17-01-08</td>
<td>✔️</td>
<td>0:05:25</td>
<td>[View pipeline]</td>
<td>-</td>
<td>3/17/2020, 1:01:58 PM</td>
<td>0.987</td>
</tr>
<tr>
<td>xgb_pipeline 2020-03-16 16-47-07</td>
<td>✔️</td>
<td>0:06:08</td>
<td>[View pipeline]</td>
<td>-</td>
<td>3/16/2020, 12:47:19 PM</td>
<td>0.983</td>
</tr>
<tr>
<td>xgb_pipeline 2020-03-16 13-57-07</td>
<td>✔️</td>
<td>0:05:18</td>
<td>[View pipeline]</td>
<td>-</td>
<td>3/15/2020, 9:57:03 AM</td>
<td>0.980</td>
</tr>
</tbody>
</table>

If you select the Snapshot step, you can see the name of the Snapshot copy that was used to run this experiment.
The described step has visual artifacts to explore the metrics we used. You can expand to view the full plot as seen in the following image.

The MLRun API database also tracks inputs, outputs, and artifacts for each run organized by project. An example of inputs, outputs, and artifacts for each run can be seen in the following image.
For each job, we store additional details.

<table>
<thead>
<tr>
<th>Name</th>
<th>Date/Time</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>deploy-model</td>
<td>24 Mar, 14:56:03</td>
<td>bce38e</td>
</tr>
<tr>
<td>xgb_train</td>
<td>24 Mar, 14:53:18</td>
<td>5e85949</td>
</tr>
<tr>
<td>data-prep</td>
<td>24 Mar, 14:52:45</td>
<td>126dc73</td>
</tr>
<tr>
<td>describe</td>
<td>24 Mar, 14:52:45</td>
<td>c2a460e</td>
</tr>
<tr>
<td>deploy-features-function</td>
<td>24 Mar, 14:52:43</td>
<td>50d8bb83</td>
</tr>
<tr>
<td>NetApp_Cloud_Volume_Snap</td>
<td>24 Mar, 14:51:22</td>
<td>3108eb2</td>
</tr>
</tbody>
</table>

There is more information about MLRun than we can cover in this document. AI artifacts, including the definition of the steps and functions, can be saved to the API database, versioned, and invoked individually or as a full project. Projects can also be saved and pushed to Git for later use. We encourage you to learn more at the MLRun GitHub site.

Deploy Grafana Dashboard

After everything is deployed, we run inferences on new data. The models predict failure on network device equipment. The results of the prediction are stored in an Iguazio TimeSeries table. You can visualize the results with Grafana in the platform integrated with Iguazio’s security and data access policy.

You can deploy the dashboard by importing the provided JSON file into the Grafana interfaces in the cluster.
1. To verify that the Grafana service is running, look under Services.

   **Services**

<table>
<thead>
<tr>
<th>Name</th>
<th>Running User</th>
<th>Version</th>
<th>CPU (cores)</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>docker-registry</td>
<td></td>
<td>2.7.1</td>
<td>96μ</td>
<td>1.67 Gi</td>
</tr>
<tr>
<td>framesd</td>
<td></td>
<td>0.5.10</td>
<td>369μ</td>
<td>795.19 MB</td>
</tr>
<tr>
<td>grafana</td>
<td></td>
<td>6.6.0</td>
<td>1m</td>
<td>38.39 MB</td>
</tr>
<tr>
<td>Jupyter</td>
<td>admin</td>
<td>1.0.2</td>
<td>81m</td>
<td>3.27 GB</td>
</tr>
<tr>
<td>Log Forwarder</td>
<td></td>
<td>6.7.2</td>
<td>0</td>
<td>0 bytes</td>
</tr>
</tbody>
</table>

2. If it is not present, deploy an instance from the Services section:
   a. Click New Service.
   b. Select Grafana from the list.
   c. Accept the defaults.
   d. Click Next Step.
   e. Enter your user ID.
   f. Click Save Service.
   g. Click Apply Changes at the top.

3. To deploy the dashboard, download the file NetopsPredictions-Dashboard.json through the Jupyter interface.
4. Open Grafana from the Services section and import the dashboard.

5. Click Upload *.json File and select the file that you downloaded earlier (NetopsPredictions-Dashboard.json). The dashboard displays after the upload is completed.
Deploy Cleanup Function

When you generate a lot of data, it is important to keep things clean and organized. To do so, deploy the cleanup function with the `cleanup.ipynb` notebook.

Benefits

NetApp and Iguazio speed up and simplify the deployment of AI and ML applications by building in essential frameworks, such as Kubeflow, Apache Spark, and TensorFlow, along with orchestration tools like Docker and Kubernetes. By unifying the end-to-end data pipeline, NetApp and Iguazio reduce the latency and complexity inherent in many advanced computing workloads, effectively bridging the gap between development and operations. Data scientists can run queries on large datasets and securely share data and algorithmic models with authorized users during the training phase. After the containerized models are ready for production, you can easily move them from development environments to operational environments.

Conclusion

When building your own AI/ML pipelines, configuring the integration, management, security, and accessibility of the components in an architecture is a challenging task. Giving developers access and control of their environment presents another set of challenges.

The combination of NetApp and Iguazio brings these technologies together as managed services to accelerate technology adoption and improve the time to market for new AI/ML applications.

**TR-4915: Data movement with E-Series and BeeGFS for AI and analytics workflows**

Cody Harryman and Ryan Rodine, NetApp

TR-4915 describes how to move data from any data repository into a BeeGFS file system
backed by NetApp E-Series SAN storage. For artificial intelligence (AI) and machine learning (ML) applications, customers might routinely need to move large data sets exceeding many petabytes of data into their BeeGFS clusters for model development. This document explores how to accomplish this by using NetApp XCP and NetApp BlueXP Copy and Sync tools.

TR-4915: Data movement with E-Series and BeeGFS for AI and analytics workflows

Use Cases

**Responsible AI and confidential inferencing - NetApp AI with Protopia Image Transformation**

**TR-4928: Responsible AI and confidential inferencing - NetApp AI with Protopia Image and Data Transformation**

Sathish Thyagarajan, Michael Oglesby, NetApp
Byung Hoon Ahn, Jennifer Cwagenberg, Protopia

Visual interpretations have become an integral part of communication with the emergence of image capturing and image processing. Artificial intelligence (AI) in digital image processing brings novel business opportunities, such as in the medical field for cancer and other disease identification, in geospatial visual analytics for studying environmental hazards, in pattern recognition, in video processing for fighting crime, and so on. However, this opportunity also comes with extraordinary responsibilities.

The more decisions organizations put into the hands of AI, the more they accept risks related to data privacy and security and legal, ethical, and regulatory issues. Responsible AI enables a practice that allows companies and government organizations to build trust and governance that is crucial for AI at scale in large enterprises. This document describes an AI inferencing solution validated by NetApp under three different scenarios by using NetApp data management technologies with Protopia data obfuscation software to privatize sensitive data and reduce risks and ethical concerns.

Millions of images are generated every day with various digital devices by both consumers and business entities. The consequent massive explosion of data and computational workload makes businesses turn to cloud computing platforms for scale and efficiency. Meanwhile, privacy concerns over the sensitive information contained in image data arise with transfer to a public cloud. The lack of security and privacy assurances become the main barrier to deployment of image-processing AI systems.

Additionally, there is the right to erasure by the GDPR, the right of an individual to request that an organization erase all their personal data. There is also the Privacy Act, which establishes a code of fair information practices. Digital images such as photographs can constitute personal data under the GDPR, which governs how data must be collected, processed, and erased. Failure to do so is a failure to comply with GDPR, which might lead to hefty fines for breaching compliances that can be seriously damaging to organizations. Privacy principles are among the backbone of implementing responsible AI that ensure fairness in the machine learning (ML) and deep learning (DL) model predictions and lowers risks associated with violating privacy or regulatory compliance.

This document describes a validated design solution under three different scenarios with and without image obfuscation relevant to preserving privacy and deploying a responsible AI solution:
• **Scenario 1.** On-demand inferencing within Jupyter notebook.
• **Scenario 2.** Batch inferencing on Kubernetes.
• **Scenario 3.** NVIDIA Triton inference server.

For this solution, we use the Face Detection Data Set and Benchmark (FDDB), a dataset of face regions designed for studying the problem of unconstrained face detection, combined with the PyTorch machine learning framework for implementation of FaceBoxes. This dataset contains the annotations for 5171 faces in a set of 2845 images of various resolutions. Furthermore, this technical report presents some of the solution areas and relevant use cases gathered from NetApp customers and field engineers in situations where this solution is applicable.

**Target audience**

This technical report is intended for the following audiences:

• Business leaders and enterprise architects who want to design and deploy responsible AI and address data protection and privacy issues concerning facial image processing in public spaces.

• Data scientists, data engineers, AI/ machine learning (ML) researchers, and developers of AI/ML systems who aim to protect and preserve privacy.

• Enterprise architects who design data obfuscation solutions for AI/ML models and applications that comply with regulatory standards such as GDPR, CCPA, or the Privacy Act of the Department of Defense (DoD) and government organizations.

• Data scientists and AI engineers looking for efficient ways to deploy deep learning (DL) and AI/ML/DL inferencing models that protect sensitive information.

• Edge device managers and edge server administrators responsible for deployment and management of edge inferencing models.

**Solution architecture**

This solution is designed to handle real-time and batch inferencing AI workloads on large datasets by using the processing power of GPUs alongside traditional CPUs. This validation demonstrates the privacy-preserving inference for ML and optimal data management required for organizations seeking responsible AI deployments. This solution provides an architecture suited for a single or multi-node Kubernetes platform for edge and cloud computing interconnected with NetApp ONTAP AI at the core on-premises, NetApp DataOps Toolkit, and Protopia obfuscation software using Jupyter Lab and CLI interfaces. The following figure shows the logical architecture overview of data fabric powered by NetApp with DataOps Toolkit and Protopia.
Protopia obfuscation software runs seamlessly on top of the NetApp DataOps Toolkit and transforms the data before leaving the storage server.

**Solution areas**

Digital image processing comes with a lot of advantages, allowing many organizations to make the most of data associated with visual representations. This NetApp and Protopia solution provides a unique AI inferencing design to protect and privatize AI/ML data across the ML/DL life cycle. It enables customers to retain ownership of sensitive data, use public- or hybrid-cloud deployment models for scale and efficiency by alleviating concerns related to privacy, and deploy AI inferencing at the edge.
Environmental intelligence

There are many ways industries can take advantage of geospatial analytics in the areas of environmental hazards. Governments and the department of public works can derive actionable insights on public health and weather conditions to better advise the public during a pandemic or a natural disaster such as wildfires. For example, you can identify a COVID-positive patient in public spaces, such as airports or hospitals, without compromising the privacy of the affected individual and alert the respective authorities and the public in the vicinity for necessary safety measures.

Edge device wearables

In the military and on battlefields, you can use AI inferencing on the edge as wearable devices to track soldier health, monitor driver behavior, and alert authorities on the safety and associated risks of approaching military vehicles while preserving and protecting the privacy of soldiers. The future of the military is going high-tech with the Internet of Battlefield Things (IoBT) and the Internet of Military Things (IoMT) for wearable combat gear that help soldiers identify enemies and perform better in battle by using rapid edge computing. Protecting and preserving visual data collected from edge devices such as drones and wearable gears is crucial to keep hackers and the enemy at bay.

Noncombatant evacuation operations

Noncombatant evacuation operations (NEOs) are conducted by the DoD to assist in evacuating US citizens and nationals, DoD civilian personnel, and designated persons (host nation (HN) and third-country nationals (TCNs)) whose lives are in danger to an appropriate safe haven. The administrative controls in place use largely manual evacuee screening processes. However, the accuracy, security, and speed of evacuee identification, evacuee tracking, and threat screening could potentially be improved by using highly automated AI/ML tools combined with AI/ML video obfuscation technologies.

Healthcare and biomedical research

Image processing is used to diagnose pathologies for surgical planning from 3D images obtained from computed tomography (CT) or magnetic resonance imaging (MRI). HIPAA privacy rules govern how data must be collected, processed, and erased by organizations for all personal information and digital images like photographs. For data to qualify as sharable under the HIPAA Safe Harbor regulations, full-face photographic images and any comparable images must be removed. Automated techniques like de-identification or skull-stripping algorithms used to obscure an individual's facial features from structural CT/MR images have become an essential part of the data sharing process for biomedical research institutions.

Cloud migration of AI/ML analytics

Enterprise customers have traditionally trained and deployed AI/ML models on-premises. For economies of scale and efficiency reasons, these customers are expanding to move AI/ML functions into public, hybrid, or multi-cloud cloud deployments. However, they are bound by what data can be exposed to other infrastructures. NetApp solutions address a full range of cybersecurity threats required for data protection and security assessment and, when combined with Protopia data transformation, minimize the risks associated with migrating image processing AI/ML workloads to the cloud.

For additional use cases for edge computing and AI inferencing across other industries, see TR-4886 AI Inferencing at the Edge and the NetApp AI blog, Intelligence versus privacy.

Technology overview

This section provides an overview of the various technical components required to complete this solution.
Protopia

Protopia AI offers a unobtrusive, software-only solution for confidential inference in the market today. The Protopia solution delivers unparalleled protection for inference services by minimizing exposure of sensitive information. AI is only fed the information in the data record that is truly essential to perform the task at hand and nothing more. Most inference tasks do not use all the information that exists in every data record. Regardless of whether your AI is consuming images, voice, video, or even structured tabular data, Protopia delivers only what the inference service needs. The patented core technology uses mathematically curated noise to stochastically transform the data and garble the information that is not needed by a given ML service. This solution does not mask the data; rather, it changes the data representation by using curated random noise.

The Protopia solution formulates the problem of changing the representation as a gradient-based perturbation maximization method that still retains the pertinent information in the input feature space with respect to the functionality of the model. This discovery process is run as a fine-tuning pass at the end of training the ML model. After the pass automatically generates a set of probability distributions, a low-overhead data transformation applies noise samples from these distributions to the data, obfuscating it before passing it to the model for inferencing.

NetApp ONTAP AI

The NetApp ONTAP AI reference architecture, powered by DGX A100 systems and NetApp cloud connected storage systems, was developed and verified by NetApp and NVIDIA. It gives IT organizations an architecture that provides the following benefits:

- Eliminates design complexities
- Allows independent scaling of compute and storage
- Enables customers to start small and scale seamlessly
- Offers a range of storage options for various performance and cost points

ONTAP AI tightly integrates DGX A100 systems and NetApp AFF A800 storage systems with state-of-the-art networking. ONTAP AI simplifies AI deployments by eliminating design complexity and guesswork. Customers can start small and grow nondisruptively while intelligently managing data from the edge to the core to the cloud and back.

The following figure shows several variations in the ONTAP AI family of solutions with DGX A100 systems. AFF A800 system performance is verified with up to eight DGX A100 systems. By adding storage controller pairs to the ONTAP cluster, the architecture can scale to multiple racks to support many DGX A100 systems and petabytes of storage capacity with linear performance. This approach offers the flexibility to alter compute-to-storage ratios independently based on the size of the DL models that are used and the required performance metrics.
For additional information about ONTAP AI, see NVA-1153: NetApp ONTAP AI with NVIDIA DGX A100 Systems and Mellanox Spectrum Ethernet Switches.

**NetApp ONTAP**

ONTAP 9.11, the latest generation of storage management software from NetApp, enables businesses to modernize infrastructure and transition to a cloud-ready data center. Leveraging industry-leading data management capabilities, ONTAP enables the management and protection of data with a single set of tools, regardless of where that data resides. You can also move data freely to wherever it is needed: the edge, the core, or the cloud. ONTAP 9.11 includes numerous features that simplify data management, accelerate, and protect critical data, and enable next generation infrastructure capabilities across hybrid cloud architectures.

**NetApp DataOps Toolkit**

NetApp DataOps Toolkit is a Python library that makes it simple for developers, data scientists, DevOps engineers, and data engineers to perform various data management tasks, such as near-instantaneous provisioning of a new data volume or JupyterLab workspace, near-instantaneous cloning of a data volume or JupyterLab workspace, and near-instantaneous taking snapshots of a data volume or JupyterLab workspace for traceability or baselining. This Python library can function as either a command-line utility or a library of functions that you can import into any Python program or Jupyter notebook.
NVIDIA Triton Inference Server

NVIDIA Triton Inference Server is an open-source inference serving software that helps standardize model deployment and execution to deliver fast and scalable AI in production. Triton Inference Server streamlines AI inferencing by enabling teams to deploy, run, and scale trained AI models from any framework on any GPU- or CPU-based infrastructure. Triton Inference Server supports all major frameworks, such as TensorFlow, NVIDIA TensorRT, PyTorch, MXNet, OpenVINO, and so on. Triton integrates with Kubernetes for orchestration and scaling that you can use in all major public cloud AI and Kubernetes platforms. It’s also integrated with many MLOps software solutions.

PyTorch

PyTorch is an open-source ML framework. It is an optimized tensor library for deep learning that uses GPUs and CPUs. The PyTorch package contains data structures for multidimensional tensors that provide many utilities for efficient serializing of tensors among other useful utilities. It also has a CUDA counterpart that enables you to run your tensor computations on an NVIDIA GPU with compute capability. In this validation, we use the OpenCV-Python (cv2) library to validate our model while taking advantage of Python’s most intuitive computer vision concepts.

Simplify data management

Data management is crucial to enterprise IT operations and data scientists so that appropriate resources are used for AI applications and training AI/ML datasets. The following additional information about NetApp technologies is out of scope for this validation but might be relevant depending on your deployment.

ONTAP data management software includes the following features to streamline and simplify operations and reduce your total cost of operation:

- Inline data compaction and expanded deduplication. Data compaction reduces wasted space inside storage blocks, and deduplication significantly increases effective capacity. This applies to data stored locally and data tiered to the cloud.
- Minimum, maximum, and adaptive quality of service (AQoS). Granular quality of service (QoS) controls help maintain performance levels for critical applications in highly shared environments.
- NetApp FabricPool. Provides automatic tiering of cold data to public and private cloud storage options, including Amazon Web Services (AWS), Azure, and NetApp StorageGRID storage solution. For more information about FabricPool, see TR-4598: FabricPool best practices.

Accelerate and protect data

ONTAP delivers superior levels of performance and data protection and extends these capabilities in the following ways:

- Performance and lower latency. ONTAP offers the highest possible throughput at the lowest possible latency.
- Data protection. ONTAP provides built-in data protection capabilities with common management across all platforms.
- NetApp Volume Encryption (NVE). ONTAP offers native volume-level encryption with both onboard and External Key Management support.
- Multitenancy and multifactor authentication. ONTAP enables sharing of infrastructure resources with the highest levels of security.
Future-proof infrastructure

ONTAP helps meet demanding and constantly changing business needs with the following features:

- Seamless scaling and nondisruptive operations. ONTAP supports the nondisruptive addition of capacity to existing controllers and to scale-out clusters. Customers can upgrade to the latest technologies, such as NVMe and 32Gb FC, without costly data migrations or outages.

- Cloud connection. ONTAP is the most cloud-connected storage management software, with options for software-defined storage (ONTAP Select) and cloud-native instances (NetApp Cloud Volumes Service) in all public clouds.

- Integration with emerging applications. ONTAP offers enterprise-grade data services for next generation platforms and applications, such as autonomous vehicles, smart cities, and Industry 4.0, by using the same infrastructure that supports existing enterprise apps.

NetApp Astra Control

The NetApp Astra product family offers storage and application-aware data management services for Kubernetes applications on-premises and in the public cloud, powered by NetApp storage and data management technologies. It enables you to easily back up Kubernetes applications, migrate data to a different cluster, and instantly create working application clones. If you need to manage Kubernetes applications running in a public cloud, see the documentation for Astra Control Service. Astra Control Service is a NetApp-managed service that provides application-aware data management of Kubernetes clusters in Google Kubernetes Engine (GKE) and Azure Kubernetes Service (AKS).

NetApp Astra Trident

Astra Trident from NetApp is an open-source dynamic storage orchestrator for Docker and Kubernetes that simplifies the creation, management, and consumption of persistent storage. Trident, a Kubernetes-native application, runs directly within a Kubernetes cluster. Trident enables customers to seamlessly deploy DL container images onto NetApp storage and provides an enterprise-grade experience for AI container deployments. Kubernetes users (ML developers, data scientists, and so on) can create, manage, and automate orchestration and cloning to take advantage of advanced data management capabilities powered by NetApp technology.

NetApp BlueXP Copy and Sync

BlueXP Copy and Sync is a NetApp service for rapid and secure data synchronization. Whether you need to transfer files between on-premises NFS or SMB file shares, NetApp StorageGRID, NetApp ONTAP S3, NetApp Cloud Volumes Service, Azure NetApp Files, Amazon Simple Storage Service (Amazon S3), Amazon Elastic File System (Amazon EFS), Azure Blob, Google Cloud Storage, or IBM Cloud Object Storage, BlueXP Copy and Sync moves the files where you need them quickly and securely. After your data is transferred, it is fully available for use on both source and target. BlueXP Copy and Sync continuously synchronizes the data based on your predefined schedule, moving only the deltas, so that time and money spent on data replication is minimized. BlueXP Copy and Sync is a software-as-a-service (SaaS) tool that is extremely simple to set up and use. Data transfers that are triggered by BlueXP Copy and Sync are carried out by data brokers. You can deploy BlueXP Copy and Sync data brokers in AWS, Azure, Google Cloud Platform, or on-premises.

NetApp BlueXP Classification

Driven by powerful AI algorithms, NetApp BlueXP Classification provides automated controls and data governance across your entire data estate. You can easily pinpoint cost-savings, identify compliance and privacy concerns, and find optimization opportunities. The BlueXP Classification dashboard gives you the insight to identify duplicate data to eliminate redundancy, map personal, nonpersonal, and sensitive data and turn on alerts for sensitive data and anomalies.
Test and validation plan

For this solution design, the following three scenarios were validated:

- An inferencing task, with and without Protopia obfuscation, within a JupyterLab workspace that was orchestrated by using the NetApp DataOps Toolkit for Kubernetes.
- A batch inferencing job, with and without Protopia obfuscation, on Kubernetes with a data volume that was orchestrated by using NetApp DataOps Toolkit for Kubernetes.
- An inferencing task using an NVIDIA Triton Inference Server instance that was orchestrated by using the NetApp DataOps Toolkit for Kubernetes. We applied Protopia obfuscation to the image before invoking the Triton inference API to simulate the common requirement that any data that is transmitted over the network must be obfuscated. This workflow is applicable to use cases where data is collected within a trusted zone but must be passed outside of that trusted zone for inferencing. Without Protopia obfuscation, it is not possible to implement this type of workflow without sensitive data leaving the trusted zone.

Test configuration

The following table outlines the solution design validation environment.

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kubernetes</td>
<td>1.21.6</td>
</tr>
<tr>
<td>NetApp Astra Trident CSI Driver</td>
<td>22.01.0</td>
</tr>
<tr>
<td>NetApp DataOps Toolkit for Kubernetes</td>
<td>2.3.0</td>
</tr>
<tr>
<td>NVIDIA Triton Inference Server</td>
<td>21.11-py3</td>
</tr>
</tbody>
</table>

Test procedure

This section describes the tasks needed to complete the validation.

Prerequisites

To execute the tasks outlined in this section, you must have access to a Linux or macOS host with the following tools installed and configured:

- Kubectl (configured for access to an existing Kubernetes cluster)
  - Installation and configuration instructions can be found [here](#).
- NetApp DataOps Toolkit for Kubernetes
  - Installation instructions can be found [here](#).

Scenario 1 – On-demand inferencing in JupyterLab

1. Create a Kubernetes namespace for AI/ML inferencing workloads.

   ```
   $ kubectl create namespace inference
   namespace/inference created
   ```

2. Use the NetApp DataOps Toolkit to provision a persistent volume for storing the data on which you will...
$ netapp_dataops_k8s_cli.py create volume --namespace=inference --pvc -name=inference-data --size=50Gi
Creating PersistentVolumeClaim (PVC) 'inference-data' in namespace 'inference'.
PersistentVolumeClaim (PVC) 'inference-data' created. Waiting for Kubernetes to bind volume to PVC.
Volume successfully created and bound to PersistentVolumeClaim (PVC) 'inference-data' in namespace 'inference'.

3. Use the NetApp DataOps Toolkit to create a new JupyterLab workspace. Mount the persistent volume that was created in the previous step by using the --mount-pvc option. Allocate NVIDIA GPUs to the workspace as necessary by using the --nvidia-gpu option.

In the following example, the persistent volume inference-data is mounted to the JupyterLab workspace container at /home/jovyan/data. When using official Project Jupyter container images, /home/jovyan is presented as the top-level directory within the JupyterLab web interface.

$ netapp_dataops_k8s_cli.py create jupyterlab --namespace=inference --workspace-name=live-inference --size=50Gi --nvidia-gpu=2 --mount-pvc=inference-data:/home/jovyan/data
Set workspace password (this password will be required in order to access the workspace):
Re-enter password:
Creating persistent volume for workspace... 
Creating PersistentVolumeClaim (PVC) 'ntap-dsutil-jupyterlab-live-inference' in namespace 'inference'.
PersistentVolumeClaim (PVC) 'ntap-dsutil-jupyterlab-live-inference' created. Waiting for Kubernetes to bind volume to PVC.
Volume successfully created and bound to PersistentVolumeClaim (PVC) 'ntap-dsutil-jupyterlab-live-inference' in namespace 'inference'.
Creating Service 'ntap-dsutil-jupyterlab-live-inference' in namespace 'inference'.
Service successfully created.
Attaching Additional PVC: 'inference-data' at mount_path: '/home/jovyan/data'.
Creating Deployment 'ntap-dsutil-jupyterlab-live-inference' in namespace 'inference'.
Deployment 'ntap-dsutil-jupyterlab-live-inference' created.
Waiting for Deployment 'ntap-dsutil-jupyterlab-live-inference' to reach Ready state.
Deployment successfully created.
Workspace successfully created.
To access workspace, navigate to http://192.168.0.152:32721
4. Access the JupyterLab workspace by using the URL specified in the output of the `create jupyterlab` command. The data directory represents the persistent volume that was mounted to the workspace.

5. Open the data directory and upload the files on which the inferencing is to be performed. When files are uploaded to the data directory, they are automatically stored on the persistent volume that was mounted to the workspace. To upload files, click the Upload Files icon, as shown in the following image.
6. Return to the top-level directory and create a new notebook.

7. Add inferencing code to the notebook. The following example shows inferencing code for an image detection use case.
8. Add Protopia obfuscation to your inferencing code. Protopia works directly with customers to provide use-case specific documentation and is outside of the scope of this technical report. The following example shows inferencing code for an image detection use case with Protopia obfuscation added.
STEP 3-2: Protopia AI (With obfuscation) detection

```python
# get current frame
frame = input_image

# preprocess input
preprocessed_input = preprocess_input(frame)
preprocessed_input = torch.tensor(preprocessed_input).to(device)

# run forward pass
net_noisy_activation = net_noisy_model.forward_head(preprocessed_input)  # run the first few layers
net_noisy_activation = net_noisy_model.forward_noise(net_noisy_activation)  # SINGLE OPTIONAL LINE FOR PRIVATE INFERENCENET
noisy_pred = noisy_model.forward_tail(noisy_activation)  # runs rest of the layers

# postprocess output
noisy_pred = [i.item() for i in noisy_pred]  # prediction
noisy_outputs = postprocess_outputs(noisy_pred, [[input_image_width, input_image_height]], priors, THRESHOLD = 0.5)

# get reconstruction of the noisy activation
noisy_reconstruction = noisy_reconstruction_decoder(noisy_activation)
noisy_reconstruction = unpreprocess_output(noisy_reconstruction)

# draw rectangles
for (x1, y1, x2, y2, s) in noisy_outputs:
    x1, y1 = int(x1), int(y1)
    x2, y2 = int(x2), int(y2)
    cv2.rectangle(noisy_reconstruction, (x1, y1), (x2, y2), (0, 0, 255), 4)
```

Visualize Protopia AI (With obfuscation) detection

```python
show_no2_image(noisy_reconstruction, scale=2)
```
Scenario 2 – Batch inferencing on Kubernetes

1. Create a Kubernetes namespace for AI/ML inferencing workloads.

```bash
$ kubectl create namespace inference
namespace/inference created
```

2. Use the NetApp DataOps Toolkit to provision a persistent volume for storing the data on which you will perform the inferencing.

```bash
$ netapp_dataops_k8s_cli.py create volume --namespace=inference --pvc -name=inference-data --size=50Gi
Creating PersistentVolumeClaim (PVC) 'inference-data' in namespace 'inference'.
PersistentVolumeClaim (PVC) 'inference-data' created. Waiting for Kubernetes to bind volume to PVC.
Volume successfully created and bound to PersistentVolumeClaim (PVC) 'inference-data' in namespace 'inference'.
```

3. Populate the new persistent volume with the data on which you will perform the inferencing.

There are several methods for loading data onto a PVC. If your data is currently stored in an S3-compatible object storage platform, such as NetApp StorageGRID or Amazon S3, then you can use NetApp DataOps Toolkit S3 Data Mover capabilities. Another simple method is to create a JupyterLab workspace and then upload files through the JupyterLab web interface, as outlined in Steps 3 to 5 in the section "Scenario 1 – On-demand inferencing in JupyterLab."

4. Create a Kubernetes job for your batch inferencing task. The following example shows a batch inferencing job for an image detection use case. This job performs inferencing on each image in a set of images and writes inferencing accuracy metrics to stdout.
$ vi inference-job-raw.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-inference-raw
  namespace: inference
spec:
  backoffLimit: 5
template:
  spec:
    volumes:
    - name: data
      persistentVolumeClaim:
        claimName: inference-data
    - name: dshm
      emptyDir:
        medium: Memory
    containers:
    - name: inference
      image: netapp-protopia-inference:latest
      imagePullPolicy: IfNotPresent
      command: ["python3", "run-accuracy-measurement.py", "--dataset", 
"/data/netapp-face-detection/FDDB"]
      resources:
        limits:
          nvidia.com/gpu: 2
      volumeMounts:
      - mountPath: /data
        name: data
      - mountPath: /dev/shm
        name: dshm
    restartPolicy: Never
$ kubectl create -f inference-job-raw.yaml
job.batch/netapp-inference-raw created

5. Confirm that the inferencing job completed successfully.
6. Add Protopia obfuscation to your inferencing job. You can find use case-specific instructions for adding Protopia obfuscation directly from Protopia, which is outside of the scope of this technical report. The following example shows a batch inferencing job for a face detection use case with Protopia obfuscation added by using an ALPHA value of 0.8. This job applies Protopia obfuscation before performing inferencing for each image in a set of images and then writes inferencing accuracy metrics to stdout.

We repeated this step for ALPHA values 0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 0.9, and 0.95. You can see the results in "Inferencing accuracy comparison."
$ vi inference-job-protopia-0.8.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-inference-protopia-0.8
  namespace: inference
spec:
  backoffLimit: 5
  template:
    spec:
      volumes:
      - name: data
        persistentVolumeClaim:
          claimName: inference-data
      - name: dshm
        emptyDir:
          medium: Memory
      containers:
      - name: inference
        image: netapp-protopia-inference:latest
        imagePullPolicy: IfNotPresent
        env:
        - name: ALPHA
          value: "0.8"
        command: ["python3", "run-accuracy-measurement.py", "--dataset", "/data/netapp-face-detection/FDDB", "--alpha", "$(ALPHA)", "--noisy"]
        resources:
          limits:
            nvidia.com/gpu: 2
        volumeMounts:
        - mountPath: /data
          name: data
        - mountPath: /dev/shm
          name: dshm
      restartPolicy: Never

$ kubectl create -f inference-job-protopia-0.8.yaml
job.batch/netapp-inference-protopia-0.8 created

7. Confirm that the inferencing job completed successfully.
Scenario 3 – NVIDIA Triton Inference Server

1. Create a Kubernetes namespace for AI/ML inferencing workloads.

   $ kubectl create namespace inference
   namespace/inference created

2. Use the NetApp DataOps Toolkit to provision a persistent volume to use as a model repository for the NVIDIA Triton Inference Server.

   $ netapp_dataops_k8s_cli.py create volume --namespace=inference --pvc
   Creating PersistentVolumeClaim (PVC) 'triton-model-repo' in namespace 'inference'.
   PersistentVolumeClaim (PVC) 'triton-model-repo' created. Waiting for Kubernetes to bind volume to PVC.
   Volume successfully created and bound to PersistentVolumeClaim (PVC) 'triton-model-repo' in namespace 'inference'.

3. Store your model on the new persistent volume in a format that is recognized by the NVIDIA Triton Inference Server.

   There are several methods for loading data onto a PVC. A simple method is to create a JupyterLab workspace and then upload files through the JupyterLab web interface, as outlined in steps 3 to 5 in “Scenario 1 – On-demand inferencing in JupyterLab.”
4. Use NetApp DataOps Toolkit to deploy a new NVIDIA Triton Inference Server instance.

```
$ netapp_dataops_k8s_cli.py create triton-server --namespace=inference
--server-name=netapp-inference --model-repo-pvc-name=triton-model-repo
Creating Service 'ntap-dsutil-triton-netapp-inference' in namespace 'inference'.
Service successfully created.
Creating Deployment 'ntap-dsutil-triton-netapp-inference' in namespace 'inference'.
Deployment 'ntap-dsutil-triton-netapp-inference' created.
Waiting for Deployment 'ntap-dsutil-triton-netapp-inference' to reach Ready state.
Deployment successfully created.
Server successfully created.
Server endpoints:
  http: 192.168.0.152:31208
  grpc: 192.168.0.152:32736
  metrics: 192.168.0.152:30009/metrics
```

5. Use a Triton client SDK to perform an inferencing task. The following Python code excerpt uses the Triton Python client SDK to perform an inferencing task for a face detection use case. This example calls the Triton API and passes in an image for inferencing. The Triton Inference Server then receives the request, invokes the model, and returns the inferencing output as part of the API results.

```
# get current frame
frame = input_image
# preprocess input
preprocessed_input = preprocess_input(frame)
preprocessed_input = torch.Tensor(preprocessed_input).to(device)
# run forward pass
clean_activation = clean_model_head(preprocessed_input)  # runs the first few layers

# pass clean image to Triton Inference Server API for inferencing
triton_client = httpclient.InferenceServerClient(url="192.168.0.152:31208",
  verbose=False)
model_name = "face_detection_base"
inputs = []
outputs = []
inputs.append(httpclient.InferInput("INPUT__0", [1, 128, 32, 32],
```
inputs[0].set_data_from_numpy(clean_activation.detach().cpu().numpy(), binary_data=False)
outputs.append(httpclient.InferRequestedOutput("OUTPUT__0", binary_data=False))
outputs.append(httpclient.InferRequestedOutput("OUTPUT__1", binary_data=False))
results = triton_client.infer(
    model_name,
    inputs,
    outputs=outputs,
    #query_params=query_params,
    headers=None,
    request_compression_algorithm=None,
    response_compression_algorithm=None)
# print(results.get_response())
statistics =
triton_client.get_inference_statistics(model_name=model_name, headers=None)
print(statistics)
if len(statistics["model_stats"]) != 1:
    print("FAILED: Inference Statistics")
sys.exit(1)
loc_numpy = results.as_numpy("OUTPUT__0")
pred_numpy = results.as_numpy("OUTPUT__1")

# postprocess output
clean_pred = (loc_numpy, pred_numpy)
clean_outputs = postprocess_outputs(
    clean_pred, [[input_image_width, input_image_height]], priors, THRESHOLD)
)
# draw rectangles
clean_frame = copy.deepcopy(frame)  # needs to be deep copy
for (x1, y1, x2, y2, s) in clean_outputs[0]:
    x1, y1 = int(x1), int(y1)
    x2, y2 = int(x2), int(y2)
    cv2.rectangle(clean_frame, (x1, y1), (x2, y2), (0, 0, 255), 4)

6. Add Protopia obfuscation to your inferencing code. You can find use case-specific instructions for adding Protopia obfuscation directly from Protopia; however, this process is outside the scope of this technical report. The following example shows the same Python code that is shown in the preceding step 5, but with Protopia obfuscation added.

Note that the Protopia obfuscation is applied to the image before it is passed to the Triton API. Thus, the
non-obfuscated image never leaves the local machine. Only the obfuscated image is passed across the network. This workflow is applicable to use cases in which data is collected within a trusted zone but then needs to be passed outside of that trusted zone for inferencing. Without Protopia obfuscation, it is not possible to implement this type of workflow without sensitive data ever leaving the trusted zone.

```python
# get current frame
frame = input_image

# preprocess input
preprocessed_input = preprocess_input(frame)
preprocessed_input = torch.Tensor(preprocessed_input).to(device)

# run forward pass
not_noisy_activation = noisy_model_head(preprocessed_input)  # runs the first few layers

# obfuscate image locally prior to inferencing
# SINGLE ADITIONAL LINE FOR PRIVATE INFERENCE
noisy_activation = noisy_model_noise(not_noisy_activation)

# pass obfuscated image to Triton Inference Server API for inferencing

triton_client = httpclient.InferenceServerClient(url="192.168.0.152:31208", verbose=False)
model_name = "face_detection_noisy"
inputs = []
outputs = []
inputs.append(httpclient.InferInput("INPUT__0", [1, 128, 32, 32], "FP32"))
inputs[0].set_data_from_numpy(noisy_activation.detach().cpu().numpy(), binary_data=False)
outputs.append(httpclient.InferRequestedOutput("OUTPUT__0", binary_data=False))
outputs.append(httpclient.InferRequestedOutput("OUTPUT__1", binary_data=False))
results = triton_client.infer(model_name, inputs, outputs=outputs, request_compression_algorithm=None)
```
response_compression_algorithm=None)
# print(results.get_response())
statistics =
triton_client.get_inference_statistics(model_name=model_name,
headers=None)
print(statistics)
if len(statistics["model_stats"]) != 1:
    print("FAILED: Inference Statistics")
sys.exit(1)

loc_numpy = results.as_numpy("OUTPUT__0")
pred_numpy = results.as_numpy("OUTPUT__1")

# postprocess output
noisy_pred = (loc_numpy, pred_numpy)
noisy_outputs = postprocess_outputs(
    noisy_pred, [[input_image_width, input_image_height]], priors,
    THRESHOLD * 0.5
)

# get reconstruction of the noisy activation
noisy_reconstruction = decoder_function(noisy_activation)
noisy_reconstruction = noisy_reconstruction.detach().cpu().numpy()[0]
noisy_reconstruction = unpreprocess_output(
    noisy_reconstruction, (input_image_width, input_image_height), True
).astype(np.uint8)

# draw rectangles
for (x1, y1, x2, y2, s) in noisy_outputs[0]:
    x1, y1 = int(x1), int(y1)
    x2, y2 = int(x2), int(y2)
    cv2.rectangle(noisy_reconstruction, (x1, y1), (x2, y2), (0, 0, 255),
4)

Inferencing accuracy comparison

For this validation, we performed inferencing for an image detection use case by using a
set of raw images. We then performed the same inferencing task on the same set of
images with Protopia obfuscation added before inferencing. We repeated the task using
different values of ALPHA for the Protopia obfuscation component. In the context of
Protopia obfuscation, the ALPHA value represents the amount of obfuscation that is
applied, with a higher ALPHA value representing a higher level of obfuscation. We then
compared inferencing accuracy across these different runs.

The following two tables provide details about our use case and outline the results.
Protopia works directly with customers to determine the appropriate ALPHA value for a specific use case.

<table>
<thead>
<tr>
<th>Component</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>FaceBoxes (PyTorch) -</td>
</tr>
<tr>
<td>Dataset</td>
<td>FDDB dataset</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protopia obfuscation</th>
<th>ALPHA</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>N/A</td>
<td>0.9337148153739079</td>
</tr>
<tr>
<td>Yes</td>
<td>0.05</td>
<td>0.9028766627325002</td>
</tr>
<tr>
<td>Yes</td>
<td>0.1</td>
<td>0.9024301009661478</td>
</tr>
<tr>
<td>Yes</td>
<td>0.2</td>
<td>0.9081836283186224</td>
</tr>
<tr>
<td>Yes</td>
<td>0.4</td>
<td>0.9073066107482036</td>
</tr>
<tr>
<td>Yes</td>
<td>0.6</td>
<td>0.8847816568680239</td>
</tr>
<tr>
<td>Yes</td>
<td>0.8</td>
<td>0.8841195749171925</td>
</tr>
<tr>
<td>Yes</td>
<td>0.9</td>
<td>0.8455427675252052</td>
</tr>
<tr>
<td>Yes</td>
<td>0.95</td>
<td>0.8455427675252052</td>
</tr>
</tbody>
</table>

**Obfuscation speed**

For this validation, we applied Protopia obfuscation to a 1920 x 1080 pixel image five times and measured the amount of time that it took for the obfuscation step to complete each time.

We used PyTorch running on a single NVIDIA V100 GPU to apply the obfuscation, and we cleared the GPU cache between runs. The obfuscation step took 5.47ms, 5.27ms, 4.54ms, 5.24ms, and 4.84ms respectively to complete across the five runs. The average speed was 5.072ms.

**Conclusion**

Data exists in three states: at rest, in transit, and in compute. An important part of any AI inferencing service should be the protection of data from threats during the entire process. Protecting data during inferencing is critical because the process can expose private information about both external customers and the business providing the inferencing service. Protopia AI is a nonobtrusive software-only solution for confidential AI inferencing in today’s market. With Protopia, AI is fed only the transformed information in the data records that is essential to carrying out the AI/ML task at hand and nothing more. This stochastic transformation is not a form of masking and is based on mathematically changing the representation of the data by using curated noise.

NetApp storage systems with ONTAP capabilities deliver the same or better performance as local SSD storage and, combined with the NetApp DataOps Toolkit, offer the following benefits to data scientists, data engineers, AI/ML developers, and business or enterprise IT decision makers:

- Effortless sharing of data between AI systems, analytics, and other critical business systems. This data
sharing reduces infrastructure overhead, improves performance, and streamlines data management across
the enterprise.

- Independently scalable compute and storage to minimize costs and improve resource usage.
- Streamlined development and deployment workflows using integrated Snapshot copies and clones for
  instantaneous and space-efficient user workspaces, integrated version control, and automated deployment.
- Enterprise-grade data protection and data governance for disaster recovery, business continuity, and
  regulatory requirements.
- Simplified invocation of data management operations; rapidly take Snapshot copies of data scientist
  workspaces for backup and traceability from the NetApp DataOps Toolkit in Jupyter notebooks.

The NetApp and Protopia solution provides a flexible, scale-out architecture that is ideal for enterprise-grade AI
inference deployments. It enables data protection and provides privacy for sensitive information where
confidential AI inferencing requirements can be met with responsible AI practices in both on-premises and
hybrid cloud deployments.

Where to find additional information and acknowledgements

To learn more about the information described in this document, refer to the following
documents and/or websites:

- NetApp ONTAP data management software — ONTAP information library
- NetApp Persistent Storage for Containers—NetApp Trident
  https://netapp.io/persistent-storage-provisioner-for-kubernetes/
- NetApp DataOps Toolkit
  https://github.com/NetApp/netapp-dataops-toolkit
- NetApp Persistent Storage for Containers—NetApp Astra Trident
  https://netapp.io/persistent-storage-provisioner-for-kubernetes/
- Protopia AI—Confidential Inference
  https://protopia.ai/blog/protopia-ai-takes-on-the-missing-link-in-ai-privacy-confidential-inference/
- NetApp BlueXP Copy and Sync
  https://docs.netapp.com/us-en/occm/concept_cloud_sync.html#how-cloud-sync-works
- NVIDIA Triton Inference Server
- NVIDIA Triton Inference Server Documentation
- FaceBoxes in PyTorch
Acknowledgments

- Mark Cates, Principal Product Manager, NetApp
- Sufian Ahmad, Technical Marketing Engineer, NetApp
- Hadi Esmaeilzadeh, Chief Technology Officer and Professor, Protopia AI

Sentiment analysis with NetApp AI

TR-4910: Sentiment Analysis from Customer Communications with NetApp AI

Rick Huang, Sathish Thyagarajan, and David Arnette, NetApp
Diego Sosa-Coba, SFL Scientific

This technical report provides design guidance for customers to perform sentiment analysis in an enterprise-level global support center by using NetApp data management technologies with an NVIDIA software framework using transfer learning and conversational AI. This solution is applicable to any industry wanting to gain customer insights from recorded speech or text files representing chat logs, emails, and other text or audio communications. We implemented an end-to-end pipeline to demonstrate automatic speech recognition, real-time sentiment analysis, and deep-learning natural-language-processing model-retraining capabilities on a GPU-accelerated compute cluster with NetApp cloud-connected all flash storage. Massive, state-of-the-art language models can be trained and optimized to perform inference rapidly with the global support center to create an exceptional customer experience and objective, long-term employee performance evaluations.

Sentiment analysis is a field of study within Natural Language Processing (NLP) by which positive, negative, or neutral sentiments are extracted from text. Conversational AI systems have risen to a near global level of integration as more and more people come to interact with them. Sentiment analysis has a variety of use cases, from determining support center employee performance in conversations with callers and providing appropriate automated chatbot responses to predicting a firm’s stock price based on the interactions between firm representatives and the audience at quarterly earnings calls. Furthermore, sentiment analysis can be used to determine the customer’s view on the products, services, or support provided by the brand.

This end-to-end solution uses NLP models to perform high level sentiment analysis that enables support-center analytical frameworks. Audio recordings are processed into written text, and sentiment is extracted from each sentence in the conversation. Results, aggregated into a dashboard, can be crafted to analyze conversation sentiments, both historically and in real-time. This solution can be generalized to other solutions with similar data modalities and output needs. With the appropriate data, other use cases can be accomplished. For example, company earnings calls can be analyzed for sentiment using the same end-to-end pipeline. Other forms of NLP analyses, such as topic modeling and named entity recognition (NER), are also possible due to the flexible nature of the pipeline.

These AI implementations were made possible by NVIDIA RIVA, the NVIDIA TAO Toolkit, and the NetApp DataOps Toolkit working together. NVIDIA’s tools are used to rapidly deploy highly performant AI solutions using prebuilt models and pipelines. The NetApp DataOps Toolkit simplifies various data management tasks to speed up development.
Customer value

Businesses see value from an employee-assessment and customer-reaction tool for text, audio, and video conversation for sentiment analysis. Managers benefit from the information presented in the dashboard, allowing for an assessment of the employees and customer satisfaction based on both sides of the conversation.

Additionally, the NetApp DataOps Toolkit manages the versioning and allocation of data within the customer's infrastructure. This leads to frequent updates of the analytics presented within the dashboard without creating unwieldy data storage costs.

Use cases

Due to the number of calls that these support centers process, assessment of call performance could take significant time if performed manually. Traditional methods, like bag-of-words counting and other methods, can achieve some automation, but these methods do not capture more nuanced aspects and semantic context of dynamic language. AI modeling techniques can be used to perform some of these more nuanced analyses in an automated manner. Furthermore, with the current state of the art, pretrained modeling tools published by NVIDIA, AWS, Google, and others, an end-to-end pipeline with complex models can be now stood up and customized with relative ease.

An end-to-end pipeline for support center sentiment analysis ingests audio files in real time as employees converse with callers. Then, these audio files are processed for use in the speech-to-text component which converts them into a text format. Each sentence in the conversation receives a label indicating the sentiment (positive, negative, or neutral).

Sentiment analysis can provide an essential aspect of the conversations for assessment of call performance. These sentiments add an additional level of depth to the interactions between employees and callers. The AI-assisted sentiment dashboard provides managers with a real-time tracking of sentiment within a conversation, along with a retrospective analysis of the employee’s past calls.

There are prebuilt tools that can be combined in powerful ways to quickly create an end-to-end AI pipeline to solve this problem. In this case, the NVIDIA RIVA library can be used to perform the two in-series tasks: audio transcription and sentiment analysis. The first is a supervised learning signal processing algorithm and the second is a supervised learning NLP classification algorithm. These out-of-the-box algorithms can be fine-tuned for any relevant use case with business-relevant data using the NVIDIA TAO Toolkit. This leads to more accurate and powerful solutions being built for only a fraction of the cost and resources. Customers can incorporate the NVIDIA Maxine framework for GPU-accelerated video conferencing applications in their support center design.

The following use cases are at the core of this solution. Both use cases use the TAO Toolkit for model fine-tuning and RIVA for model deployment.

- Speech-to-text
- Sentiment analysis

To analyze support center interactions between employees and customers, each customer conversation in the form of audio calls can be run through the pipeline to extract sentence-level sentiments. Those sentiments can then be verified by a human to justify the sentiments or adjust them as needed. The labeled data is then passed onto the fine-tuning step to improve sentiment predictions. If labeled sentiment data already exists, then model fine-tuning can be expedited. In either case, the pipeline is generalizable to other solutions that require the ingestion of audio and the classification of sentences.
AI sentiment outputs are either uploaded to an external cloud database or to a company-managed storage system. The sentiment outputs are transferred from this larger database into local storage for use within the dashboard that displays the sentiment analysis for managers. The dashboard’s primary functionality is to interface with the customer service employee in real time. Managers can assess and provide feedback on employees during their calls with live updates of the sentiment of each sentence, as well as an historic review of the employee’s past performance or customer reactions.

The NetApp DataOps Toolkit can continue to manage data storage systems even after the RIVA inference pipeline generates sentiment labels. Those AI results can be uploaded to a data storage system managed by the NetApp DataOps Toolkit. The data storage systems must be capable of managing hundreds of inserts and selects every minute. The local device storage system queries the larger data storage in real-time for extraction. The larger data storage instance can also be queried for historical data to further enhance the dashboard experience. The NetApp DataOps Toolkit facilitates both these uses by rapidly cloning data and distributing it across all the dashboards that use it.
Target Audience

The target audience for the solution includes the following groups:

- Employee managers
- Data engineers/data scientists
- IT administrators (on-premises, cloud, or hybrid)

Tracking sentiments throughout conversations is a valuable tool for assessing employee performance. Using the AI-dashboard, managers can see how employees and callers change their feelings in real time, allowing for live assessments and guidance sessions. Moreover, businesses can gain valuable customer insights from customers engaged in vocal conversations, text chatbots, and video conferencing. Such customer analytics uses the capabilities of multimodal processing at scale with modern, state-of-the-art AI models and workflows.

On the data side, a large number of audio files are processed daily by the support center. The NetApp DataOps Toolkit facilitates this data handling task for both the periodic fine-tuning of models and sentiment analysis dashboards.

IT administrators also benefit from the NetApp DataOps Toolkit as it allows them to move data quickly between deployment and production environments. The NVIDIA environments and servers must also be managed and distributed to allow for real time inference.

Architecture

The architecture of this support center solution revolves around NVIDIA’s prebuilt tools and the NetApp DataOps Toolkit. NVIDIA’s tools are used to rapidly deploy high-performance AI-solutions using prebuilt models and pipelines. The NetApp DataOps Toolkit simplifies various data management tasks to speed up development.

Solution technology

**NVIDIA RIVA** is a GPU-accelerated SDK for building multimodal conversational AI applications that deliver real-time performance on GPUs. The NVIDIA Train, Adapt, and Optimize (TAO) Toolkit provides a faster, easier way to accelerate training and quickly create highly accurate and performant, domain-specific AI models.

The NetApp DataOps Toolkit is a Python library that makes it simple for developers, data scientists, DevOps engineers, and data engineers to perform various data management tasks. This includes near-instantaneous provisioning of a new data volume or JupyterLab workspace, near-instantaneous cloning of a data volume or JupyterLab workspace, and near-instantaneous snapshotting of a data volume or JupyterLab workspace for traceability and baselining.

Architectural Diagram

The following diagram shows the solution architecture. There are three main environment categories: the cloud, the core, and the edge. Each of the categories can be geographically dispersed. For example, the cloud contains object stores with audio files in buckets in different regions, whereas the core might contain datacenters linked via a high-speed network or NetApp BlueXP Copy and Sync. The edge nodes denote the individual human agent’s daily working platforms, where interactive dashboard tools and microphones are available to visualize sentiment and collect audio data from conversations with customers.

In GPU-accelerated datacenters, businesses can use the NVIDIA RIVA framework to build conversational AI applications, to which the Tao Toolkit connects for model finetuning and retraining using transfer L-learning techniques. These compute applications and workflows are powered by the NetApp DataOps Toolkit, enabling the best data management capabilities ONTAP has to offer. The toolkit allows corporate data teams to rapidly
prototype their models with associated structured and unstructured data via snapshots and clones for traceability, versioning, A/B testing, thus providing security, governance, and regulatory compliance. See the section "Storage Design" for more details.

This solution demonstrates the audio file processing, NLP model training, transfer learning, and data management detail steps. The resulting end-to-end pipeline generates a sentiment summary that displays in real-time on human support agents’ dashboards.

Hardware requirements

The following table lists the hardware components that are required to implement the solution. The hardware components that are used in any particular implementation of the solution might vary based on customer requirements.

<table>
<thead>
<tr>
<th>Response latency tests</th>
<th>Time (milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data processing</td>
<td>10</td>
</tr>
<tr>
<td>Inferencing</td>
<td>10</td>
</tr>
</tbody>
</table>

These response-time tests were run on 50,000+ audio files across 560 conversations. Each audio file was ~100KB in size as an MP3 and ~1 MB when converted to WAV. The data processing step converts MP3s into WAV files. The inference steps convert the audio files into text and extract a sentiment from the text. These steps are all independent of one another and can be parallelized to speed up the process.

Taking into account the latency of transferring data between stores, managers should be able to see updates to the real time sentiment analysis within a second of the end of the sentence.
### NVIDIA RIVA hardware

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>Linux x86_64</td>
</tr>
<tr>
<td>GPU memory (ASR)</td>
<td>Streaming models: ~5600 MB</td>
</tr>
<tr>
<td></td>
<td>Non-streaming models: ~3100 MB</td>
</tr>
<tr>
<td>GPU memory (NLP)</td>
<td>~500MB per BERT model</td>
</tr>
</tbody>
</table>

### NVIDIA TAO Toolkit hardware

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>System RAM</td>
<td>32GB</td>
</tr>
<tr>
<td>GPU RAM</td>
<td>32GB</td>
</tr>
<tr>
<td>CPU</td>
<td>8 core</td>
</tr>
<tr>
<td>GPU</td>
<td>NVIDIA (A100, V100 and RTX 30x0)</td>
</tr>
<tr>
<td>SSD</td>
<td>100GB</td>
</tr>
</tbody>
</table>

### Flash storage system

#### NetApp ONTAP 9

ONTAP 9.9, the latest generation of storage management software from NetApp, enables businesses to modernize infrastructure and transition to a cloud-ready data center. Leveraging industry-leading data management capabilities, ONTAP enables the management and protection of data with a single set of tools, regardless of where that data resides. You can also move data freely to wherever it is needed: the edge, the core, or the cloud. ONTAP 9.9 includes numerous features that simplify data management, accelerate, and protect critical data, and enable next generation infrastructure capabilities across hybrid cloud architectures.

#### NetApp BlueXP Copy and Sync

**BlueXP Copy and Sync** is a NetApp service for rapid and secure data synchronization that allows you to transfer files between on-premises NFS or SMB file shares to any of the following targets:

- NetApp StorageGRID
- NetApp ONTAP S3
- NetApp Cloud Volumes Service
- Azure NetApp Files
- Amazon Simple Storage Service (Amazon S3)
- Amazon Elastic File System (Amazon EFS)
- Azure Blob
- Google Cloud Storage
- IBM Cloud Object Storage

BlueXP Copy and Sync moves the files where you need them quickly and securely. After your data is transferred, it is fully available for use on both the source and the target. BlueXP Copy and Sync continuously...
synchronizes the data, based on your predefined schedule, moving only the deltas, so that time and money spent on data replication is minimized. BlueXP Copy and Sync is a software as a service (SaaS) tool that is simple to set up and use. Data transfers that are triggered by BlueXP Copy and Sync are carried out by data brokers. You can deploy BlueXP Copy and Sync data brokers in AWS, Azure, Google Cloud Platform, or on-premises.

**NetApp StorageGRID**

The StorageGRID software-defined object storage suite supports a wide range of use cases across public, private, and hybrid multi-cloud environments seamlessly. With industry leading innovations, NetApp StorageGRID stores, secures, protect, and preserves unstructured data for multi-purpose use including automated lifecycle management for long periods of time. For more information, see the NetApp StorageGRID site.

**Software requirements**

The following table lists the software components that are required to implement this solution. The software components that are used in any particular implementation of the solution might vary based on customer requirements.

<table>
<thead>
<tr>
<th>Host machine</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIVA (formerly JARVIS)</td>
<td>1.4.0</td>
</tr>
<tr>
<td>TAO Toolkit (formerly Transfer Learning Toolkit)</td>
<td>3.0</td>
</tr>
<tr>
<td>ONTAP</td>
<td>9.9.1</td>
</tr>
<tr>
<td>DGX OS</td>
<td>5.1</td>
</tr>
<tr>
<td>DOTK</td>
<td>2.0.0</td>
</tr>
</tbody>
</table>

**NVIDIA RIVA Software**

<table>
<thead>
<tr>
<th>Software</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Docker</td>
<td>&gt;19.02 (with nvidia-docker installed) &gt;=19.03 if not using DGX</td>
</tr>
<tr>
<td>NVIDIA Driver</td>
<td>465.19.01+ 418.40+, 440.33+, 450.51+, 460.27+ for Data Center GPUs</td>
</tr>
<tr>
<td>Container OS</td>
<td>Ubuntu 20.04</td>
</tr>
<tr>
<td>CUDA</td>
<td>11.3.0</td>
</tr>
<tr>
<td>cuBLAS</td>
<td>11.5.1.101</td>
</tr>
<tr>
<td>cuDNN</td>
<td>8.2.0.41</td>
</tr>
<tr>
<td>NCCL</td>
<td>2.9.6</td>
</tr>
<tr>
<td>TensorRT</td>
<td>7.2.3.4</td>
</tr>
<tr>
<td>Triton Inference Server</td>
<td>2.9.0</td>
</tr>
</tbody>
</table>
NVIDIA TAO Toolkit software

<table>
<thead>
<tr>
<th>Software</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ubuntu 18.04 LTS</td>
<td>18.04</td>
</tr>
<tr>
<td>python</td>
<td>&gt;=3.6.9</td>
</tr>
<tr>
<td>docker-ce</td>
<td>&gt;19.03.5</td>
</tr>
<tr>
<td>docker-API</td>
<td>1.40</td>
</tr>
<tr>
<td>nvidia-container-toolkit</td>
<td>&gt;1.3.0-1</td>
</tr>
<tr>
<td>nvidia-container-runtime</td>
<td>3.4.0-1</td>
</tr>
<tr>
<td>nvidia-docker2</td>
<td>2.5.0-1</td>
</tr>
<tr>
<td>nvidia-driver</td>
<td>&gt;455</td>
</tr>
<tr>
<td>python-pip</td>
<td>&gt;21.06</td>
</tr>
<tr>
<td>nvidia-pyindex</td>
<td>Latest version</td>
</tr>
</tbody>
</table>

Use case details

This solution applies to the following use cases:

- Speech-to-text
- Sentiment analysis
The speech-to-text use case begins by ingesting audio files for the support centers. This audio is then processed to fit the structure required by RIVA. If the audio files have not already been split into their units of analysis, then this must be done before passing the audio to RIVA. After the audio file is processed, it is passed to the RIVA server as an API call. The server employs one of the many models it is hosting and returns a response. This speech-to-text (part of Automatic Speech Recognition) returns a text representation of the audio. From there, the pipeline switches over to the sentiment analysis portion.

For sentiment analysis, the text output from the Automatic Speech Recognition serves as the input to the Text Classification. Text Classification is the NVIDIA component for classifying text to any number of categories. The sentiment categories range from positive to negative for the support center conversations. The performance of the models can be assessed using a holdout set to determine the success of the fine-tuning step.
A similar pipeline is used for both the speech-to-text and sentiment analysis within the TAO Toolkit. The major difference is the use of labels which are required for the fine-tuning of the models. The TAO Toolkit pipeline begins with the processing of the data files. Then the pretrained models (coming from the NVIDIA NGC Catalog) are fine-tuned using the support center data. The fine-tuned models are evaluated based on their corresponding performance metrics and, if they are more performant than the pretrained models, are deployed to the RIVA server.

Design considerations

This section describes the design considerations for the different components of this solution.

Network and compute design

Depending on the restrictions on data security, all data must remain within the customer's infrastructure or a secure environment.
Storage design

The NetApp DataOps Toolkit serves as the primary service for managing storage systems. The DataOps Toolkit is a Python library that makes it simple for developers, data scientists, DevOps engineers, and data engineers to perform various data management tasks, such as near-instantaneous provisioning of a new data volume or JupyterLab workspace, near-instantaneous cloning of a data volume or JupyterLab workspace, and near-instantaneous snapshotting of a data volume or JupyterLab workspace for traceability or baselining. This Python library can function as either a command line utility or a library of functions that can be imported into any Python program or Jupyter Notebook.

RIVA best practices

NVIDIA provides several general best data practices for using RIVA:

- **Use lossless audio formats if possible.** The use of lossy codecs such as MP3 can reduce quality.
- **Augment training data.** Adding background noise to audio training data can initially decrease accuracy and yet increase robustness.
- **Limit vocabulary size if using scraped text.** Many online sources contain typos or ancillary pronouns and uncommon words. Removing these can improve the language model.
- **Use a minimum sampling rate of 16kHz if possible.** However, try not to resample, because doing so decreases audio quality.

In addition to these best practices, customers must prioritize gathering a representative sample dataset with accurate labels for each step of the pipeline. In other words, the sample dataset should proportionally reflect specified characteristics exemplified in a target dataset. Similarly, the dataset annotators have a responsibility to balance accuracy and the speed of labeling so that the quality and quantity of the data are both maximized. For example, this support center solution requires audio files, labeled text, and sentiment labels. The sequential nature of this solution means that errors from the beginning of the pipeline are propagated all the way through to the end. If the audio files are of poor quality, the text transcriptions and sentiment labels will be as well.

This error propagation similarly applies to the models trained on this data. If the sentiment predictions are 100% accurate but the speech-to-text model performs poorly, then the final pipeline is limited by the initial audio-to-text transcriptions. It is essential that developers consider each model’s performance individually and as a component of a larger pipeline. In this particular case, the end goal is to develop a pipeline that can accurately predict the sentiment. Therefore, the overall metric on which to assess the pipeline is the accuracy of the sentiments, which the speech-to-text transcription directly affects.
The NetApp DataOps Toolkit complements the data quality-checking pipeline through the use of its near-instantaneous data cloning technology. Each labeled file must be assessed and compared to the existing labeled files. Distributing these quality checks across various data storage systems ensures that these checks are executed quickly and efficiently.

**Deploying support center sentiment analysis**

Deploying the solution involves the following components:

1. NetApp DataOps Toolkit
2. NGC Configuration
3. NVIDIA RIVA Server
4. NVIDIA TAO Toolkit
5. Export TAO models to RIVA

To perform deployment, complete the following steps:

**NetApp DataOps Toolkit: Support center sentiment analysis**

To use the NetApp DataOps Toolkit, complete the following steps:

1. Pip install the toolkit.
   ```
   python3 -m pip install netapp-dataops-traditional
   ```
2. Configure the data management
   ```
   netapp_dataops_cli.py config
   ```
NGC configuration: Support center sentiment analysis

To set up **NVIDIA NGC**, complete the following steps:

1. Download the NGC.

```bash
```

2. Add your current directory to path.

```bash
echo "export PATH="\$PATH\$(pwd)"" >> ~/.bash_profile && source ~/.bash_profile
```

3. You must configure NGC CLI for your use so that you can run the commands. Enter the following command, including your API key when prompted.

```bash
ngc config set
```

For operating systems that are not Linux-based, visit [here](#).

NVIDIA RIVA server: Support center sentiment analysis

To set up **NVIDIA RIVA**, complete the following steps:

1. Download the RIVA files from NGC.

```bash
ngc registry resource download-version nvidia/riva/riva_quickstart:1.4.0-beta
```

2. Initialize the RIVA setup (`riva_init.sh`).

3. Start the RIVA server (`riva_start.sh`).

4. Start the RIVA client (`riva_start_client.sh`).

5. Within the RIVA client, install the audio processing library ([FFMPEG](#))

```bash
apt-get install ffmpeg
```

6. Start the **Jupyter** server.

7. Run the RIVA Inference Pipeline Notebook.
NVIDIA TAO Toolkit: Support center sentiment analysis

To set up NVIDIA TAO Toolkit, complete the following steps:

1. Prepare and activate a virtual environment for TAO Toolkit.
2. Install the required packages.
3. Manually pull the image used during training and fine-tuning.

   docker pull nvcr.io/nvidia/tao/tao-toolkit-pyt:v3.21.08-py3

4. Start the Jupyter server.
5. Run the TAO Fine-Tuning Notebook.

Export TAO models to RIVA: Support center sentiment analysis

To use TAO Toolkit models in RIVA, complete the following steps:

1. Save models within the TAO Fine-Tuning Notebook.
2. Copy TAO trained models to the RIVA model directory.
3. Start the RIVA server (riva_start.sh).

Deployment roadblocks

Here are a few things to keep in mind as you develop your own solution:

- The NetApp DataOps Toolkit is installed first to ensure that the data storage system runs optimally.
- NVIDIA NGC must be installed before anything else because it authenticates the downloading of images and models.
- RIVA must be installed before the TAO Toolkit. The RIVA installation configures the docker daemon to pull images as needed.
- DGX and docker must have internet access to download the models.

Validation results

As mentioned in the previous section, errors are propagated throughout the pipeline whenever there are two or more machine learning models running in sequence. For this solution, the sentiment of the sentence is the most important factor in measuring the firm’s stock risk level. The speech-to-text model, although essential to the pipeline, serves as the preprocessing unit before the sentiments can be predicted. What really matters is the difference in sentiment between the ground truth sentences and the predicted sentences. This serves as a proxy for the word error rate (WER). The speech-to-text accuracy is important, but the WER is not directly used in the final pipeline metric.

   PIPELINE_SENTIMENT_METRIC = MEAN(DIFF(GT_sentiment, ASR_sentiment))

These sentiment metrics can be calculated for the F1 Score, Recall, and Precision of each sentence. The
results can be then aggregated and displayed within a confusion matrix, along with the confidence intervals for each metric.

The benefit of using transfer learning is an increase in model performance for a fraction of data requirements, training time, and cost. The fine-tuned models should also be compared to their baseline versions to ensure the transfer learning enhances the performance instead of impairing it. In other words, the fine-tuned model should perform better on the support center data than the pretrained model.

### Pipeline assessment

<table>
<thead>
<tr>
<th>Test case</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test number</td>
<td>Pipeline sentiment metric</td>
</tr>
<tr>
<td>Test prerequisites</td>
<td>Fine-tuned models for speech-to-text and sentiment analysis models</td>
</tr>
<tr>
<td>Expected outcome</td>
<td>The sentiment metric of the fine-tuned model performs better than the original pretrained model.</td>
</tr>
</tbody>
</table>

### Pipeline sentiment metric

1. Calculate the sentiment metric for the baseline model.
2. Calculate the sentiment metric for the fine-tuned model.
3. Calculate the difference between those metrics.
4. Average the differences across all sentences.

### Videos and demos

There are two notebooks that contain the sentiment analysis pipeline: “Support-Center-Model-Transfer-Learning-and-Fine-Tuning.ipynb” and “Support-Center-Sentiment-Analysis-Pipeline.ipynb”. Together, these notebooks demonstrate how to develop a pipeline to ingest support center data and extract sentiments from each sentence using state-of-the-art deep learning models fine-tuned on the user’s data.

**Support Center - Sentiment Analysis Pipeline.ipynb**

This notebook contains the inference RIVA pipeline for ingesting audio, converting it to text, and extracting sentiments for use in an external dashboard. Dataset are automatically downloaded and processed if this has not already been done. The first section in the notebook is the Speech-to-Text which handles the conversion of audio files to text. This is followed by the Sentiment Analysis section which extracts sentiments for each text sentence and displays those results in a format similar to the proposed dashboard.

This notebook must be run before the model training and fine-tuning because the MP3 dataset must be downloaded and converted into the correct format.
Call Center - Sentiment Analysis Pipeline

This notebook demonstrates how to build a pipeline for sentiment analysis of call center conversations. The goal of this pipeline is to develop sentiment analysis for use within an external dashboard.

This tutorial will guide you through the use of NVIDIA’s RIVA for automatic speech recognition and text classification. This tutorial uses NetApp cloud storage for data storage and a pre-trained RIVA model.

Channels

These are the channels on which RIVA is hosting models.

- **speech**: 51051
- **voice**: 61051

These channels must be aligned with `riva_speech_api_port` and `riva_vision_api_port` within `config.sh`.

```python
In [4]: speech_channel = "localhost:51051"
voice_channel = "localhost:61051"
```

Speech-To-Text

Automatic Speech Recognition (ASR) takes as input an audio stream or audio buffer and returns one or more text transcripts, along with additional optional metadata. ASR represents a full speech recognition pipeline that is GPU accelerated with optimized performance and accuracy. ASR supports synchronous and streaming recognition modes.

For more information on NVIDIA RIVA’s Automatic Speech Recognition, visit [here](#).

Constants

Use these constants to affect different aspects of this pipeline:

- `DATA_DIR`: base folder where data is stored
- `DATASET_NAME`: name of the call center dataset
- `COMPANY_DATE`: folder name identifying the particular call center conversation

Support Center - Model Training and Fine-Tuning.ipynb

The TAO Toolkit virtual environment must be set up before executing the notebook (see the TAO Toolkit section in the Commands Overview for installation instructions).

This notebook relies on the TAO Toolkit to fine-tune deep learning models on the customers data. As with the previous notebook, this one is separated into two sections for the Speech-to-Text and Sentiment Analysis components. Each section goes through data processing, model training and fine-tuning, evaluation of results, and model export. Finally, there is an end section for deploying both your fine-tuned models for use in RIVA.
Call Center - Model Transfer Learning and Fine-Tuning

TAO Toolkit is a python based AI toolkit for taking purpose-built pre-trained AI models and customizing them with your own data. Transfer learning extracts learned features from an existing neural network to a new one. Transfer learning is often used when creating a large training dataset is not feasible in order to enhance the base performance of state-of-the-art models.

For this call center solution, the speech-to-text and sentiment analysis models are fine-tuned on call center data to augment the model performance on business specific terminology.

For more information on the TAO Toolkit, please visit here.

Installing necessary dependencies

For ease of use, please install TAO Toolkit inside a python virtual environment. We recommend performing this step first and then launching the notebook from the virtual environment. Please refer to the README for these instructions.

Conclusion

As customer experience has become increasingly regarded as a key competitive battleground, an AI-augmented global support center becomes a critical component that companies in almost every industry cannot afford to neglect. The solution proposed in this technical report has been demonstrated to support the delivery of such exceptional customer experiences, and the challenge now is to ensure businesses are taking actions to modernize their AI infrastructure and workflows.

The best implementations of AI in customer service are not to replace human agents. Rather, AI can empower them to create exceptional customer experiences via real-time sentiment analysis, dispute escalation, and multimodal affective computing to detect verbal, non-verbal, and facial cues with which comprehensive AI models can make recommendations at scale and supplement what an individual human agent might be
lacking. AI can also provide a better match between a particular customer with currently available agents. Using AI, businesses can extract valuable customer sentiment regarding their thoughts and impressions of the provider’s products, services, and brand image.

The solution can also be used to construct time-series data for support agents to serve as an objective performance evaluation metric. Conventional customer satisfaction surveys often lack sufficient responses. By collecting long-term employee and customer sentiment, employers can make informed decisions regarding support agents’ performance.

The combination of NetApp, SFL Scientific, opens-source orchestration frameworks, and NVIDIA brings the latest technologies together as managed services with great flexibility to accelerate technology adoption and improve the time to market for new AI/ML applications. These advanced services are delivered on-premises that can be easily ported for cloud-native environment as well as hybrid deployment architectures.

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

- 3D interactive demos
  www.netapp.com/ai
- Connect directly with a NetApp AI specialist
  https://www.netapp.com/artificial-intelligence/
- NVIDIA Base Command Platform with NetApp solution brief
- NetApp for AI 10 Good Reasons infographic
- AI in Healthcare: Deep learning to identify COVID-19 lesions in lung CT scans white paper
- AI in Healthcare: Monitoring face mask usage in healthcare settings white paper
- AI in Healthcare: Diagnostic Imaging Technical Report
- AI for Retail: NetApp Conversational AI using NVIDIA RIVA
- NetApp ONTAP AI solution brief
Distributed training in Azure - Click-Through Rate Prediction

TR-4904: Distributed training in Azure - Click-Through Rate Prediction

Rick Huang, Verron Martina, Muneer Ahmad, NetApp

The work of a data scientist should be focused on the training and tuning of machine learning (ML) and artificial intelligence (AI) models. However, according to research by Google, data scientists spend approximately 80% of their time figuring out how to make their models work with enterprise applications and run at scale.

To manage end-to-end AI/ML projects, a wider understanding of enterprise components is needed. Although DevOps have taken over the definition, integration, and deployment, these types of components, ML operations target a similar flow that includes AI/ML projects. To get an idea of what an end-to-end AI/ML pipeline touches in the enterprise, see the following list of required components:

- Storage
- Networking
- Databases
- File systems
- Containers
• Continuous integration and continuous deployment (CI/CD) pipeline
• Integrated development environment (IDE)
• Security
• Data access policies
• Hardware
• Cloud
• Virtualization
• Data science toolsets and libraries

Target audience

The world of data science touches multiple disciplines in IT and business:

• The data scientist needs the flexibility to use their tools and libraries of choice.
• The data engineer needs to know how the data flows and where it resides.
• A DevOps engineer needs the tools to integrate new AI/ML applications into their CI/CD pipelines.
• Cloud administrators and architects need to be able to set up and manage Azure resources.
• Business users want to have access to AI/ML applications.

In this technical report, we describe how Azure NetApp Files, RAPIDS AI, Dask, and Azure help each of these roles bring value to business.

Solution overview

This solution follows the lifecycle of an AI/ML application. We start with the work of data scientists to define the different steps needed to prepare data and train models. By leveraging RAPIDS on Dask, we perform distributed training across the Azure Kubernetes Service (AKS) cluster to drastically reduce the training time when compared to the conventional Python scikit-learn approach. To complete the full cycle, we integrate the pipeline with Azure NetApp Files.

Azure NetApp Files provides various performance tiers. Customers can start with a Standard tier and scale out and scale up to a high-performance tier nondisruptively without moving any data. This capability enables data scientists to train models at scale without any performance issues, avoiding any data silos across the cluster, as shown in figure below.
Technology overview

This page provides an overview of the technology used in this solution.

Microsoft and NetApp

Since May 2019, Microsoft has delivered an Azure native, first-party portal service for enterprise NFS and SMB file services based on NetApp ONTAP technology. This development is driven by a strategic partnership between Microsoft and NetApp and further extends the reach of world-class ONTAP data services to Azure.

Azure NetApp Files

The Azure NetApp Files service is an enterprise-class, high-performance, metered file storage service. Azure NetApp Files supports any workload type and is highly available by default. You can select service and performance levels and set up Snapshot copies through the service. Azure NetApp Files is an Azure first-party service for migrating and running the most demanding enterprise-file workloads in the cloud, including databases, SAP, and high-performance computing applications with no code changes.

This reference architecture gives IT organizations the following advantages:

- Eliminates design complexities
- Enables independent scaling of compute and storage
- Enables customers to start small and scale seamlessly
- Offers a range of storage tiers for various performance and cost points

Dask and NVIDIA RAPIDS overview

Dask is an open-source, parallel computing tool that scales Python libraries on multiple machines and provides faster processing of large amounts of data. It provides an API similar to single-threaded conventional Python libraries, such as Pandas, Numpy, and scikit-learn. As a result, native Python users are not forced to change much in their existing code to use resources across the cluster.

NVIDIA RAPIDS is a suite of open-source libraries that makes it possible to run end-to-end ML and data analytics workflows entirely on GPUs. Together with Dask, it enables you to easily scale from GPU workstation (scale up) to multinode, multi-GPU clusters (scale out).
For deploying Dask on a cluster, you could use Kubernetes for resource orchestration. You could also scale up or scale down the worker nodes as per the process requirement, which in-turn can help to optimize the cluster resource consumption, as shown in the following figure.

Software requirements

The following table lists the software requirements needed for this solution.

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azure Kubernetes Service</td>
<td>1.18.14</td>
</tr>
<tr>
<td>RAPIDS and Dask container image</td>
<td>Repository: &quot;rapidsai/rapidsai&quot; Tag: 0.17-cuda11.0-runtime-ubuntu18.04</td>
</tr>
<tr>
<td>NetApp Trident</td>
<td>20.01.1</td>
</tr>
<tr>
<td>Helm</td>
<td>3.0.0</td>
</tr>
</tbody>
</table>

Cloud resource requirements

This page describes the configuration of cloud resources for Azure NetApp Files.

Configure Azure NetApp Files

Configure Azure NetApp Files as described in QuickStart: Set up Azure NetApp Files and create an NFS volume.

You can proceed past the section “Create NFS volume for Azure NetApp Files” because you are going to create volumes through Trident. Before continuing, complete the following steps:

1. Register for Azure NetApp Files and NetApp Resource Provider (through the Azure Shell) (link).
2. Create an account in Azure NetApp Files (link).

3. Set up a capacity pool (a minimum 4TB Standard or Premium, depending on your need) (link). The following table lists the network configuration requirements for setting up in the cloud. The Dask cluster and Azure NetApp Files must be in the same Azure Virtual Network (VNet) or a peered VNet.

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type/version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azure Kubernetes Service</td>
<td>1.18.14</td>
</tr>
<tr>
<td>Agent node</td>
<td>3x Standard_DS2_v2</td>
</tr>
<tr>
<td>GPU node</td>
<td>3x Standard_NC6s_v3</td>
</tr>
<tr>
<td>Azure NetApp Files</td>
<td>Standard capacity pool</td>
</tr>
<tr>
<td>Capacity in TB</td>
<td>4</td>
</tr>
</tbody>
</table>

Click-through rate prediction use case summary

This use case is based on the publicly available Terabyte Click Logs dataset from Criteo AI Lab. With the recent advances in ML platforms and applications, a lot of attention is now on learning at scale. The click-through rate (CTR) is defined as the average number of click-throughs per hundred online ad impressions (expressed as a percentage). It is widely adopted as a key metric in various industry verticals and use cases, including digital marketing, retail, e-commerce, and service providers. Examples of using CTR as an important metric for potential customer traffic include the following:

• **Digital marketing:** In Google Analytics, CTR can be used to gauge how well an advertiser or merchant’s keywords, ads, and free listings are performing. A high CTR is a good indication that users find your ads and listings helpful and relevant. CTR also contributes to your keyword’s expected CTR, which is a component of Ad Rank.

• **E-commerce:** In addition to leveraging Google Analytics, there are at least some visitor statistics in an e-commerce backend. Although these statistics might not seem useful at first glance, they are typically easy to read and might be more accurate than other information. First-party datasets composed of such statistics are proprietary and are therefore the most relevant to e-commerce sellers, buyers, and platforms. These datasets can be used for setting benchmarks, comparing results to last year and yesterday by constructing a time-series for further analysis.

• **Retail:** Brick-and-mortar retailers can correlate the number of visitors and the number of customers to the CTR. The number of customers can be seen from their point-of-sale history. The CTR from retailers' websites or ad traffic might result in the aforementioned sales. Loyalty programs are another use case, because customers redirected from online ads or other websites might join to earn rewards. Retailers can acquire customers via loyalty programs and record behaviors from sales histories to build a recommendation system that not only predicts consumer buying behaviors in different categories but also personalizes coupons and decreases churn.

• **Service providers:** Telecommunication companies and internet service providers have an abundance of first-party user telemetry data for insightful AI, ML, and analytics use cases. For example, a telecom can leverage its mobile subscribers' web browsing top level domain history logs daily to fine-tune existing models to produce up-to-date audience segmentation, predict customer behavior, and collaborate with advertisers to place real-time ads for better online experience. In such data-driven marketing workflow, CTR is an important metric to reflect conversions.

In the context of digital marketing, Criteo Terabyte Click Logs are now the dataset of reference in assessing the scalability of ML platforms and algorithms. By predicting the click-through rate, an advertiser can select the
visitors who are most likely to respond to the ads, analyze their browsing history, and show the most relevant ads based on the interests of the user.

The solution provided in this technical report highlights the following benefits:

- Azure NetApp Files advantages in distributed or large-scale training
- RAPIDS CUDA-enabled data processing (cuDF, cuPy, and so on) and ML algorithms (cuML)
- The Dask parallel computing framework for distributed training

An end-to-end workflow built on RAPIDS AI and Azure NetApp Files demonstrates the drastic improvement in random forest model training time by two orders of magnitude. This improvement is significant comparing to the conventional Pandas approach when dealing with real-world click logs with 45GB of structured tabular data (on average) each day. This is equivalent to a DataFrame containing roughly twenty billion rows. We will demonstrate cluster environment setup, framework and library installation, data loading and processing, conventional versus distributed training, visualization and monitoring, and compare critical end-to-end runtime results in this technical report.

Setup

Install and set up the AKS cluster

To install and set up the AKS cluster, see the webpage [Create an AKS Cluster](#) and then complete the following steps:

1. When selecting the type of node (system [CPU] or worker [GPU] nodes), select the following:
   a. Primary system nodes should be Standard DS2v2 (agentpool default three nodes).
   b. Then add the worker node Standard_NC6s_v3 pool (three nodes minimum) for the user group (for GPU nodes) named gpupool.

2. Deployment takes 5 to 10 minutes. After it is complete, click Connect to Cluster.
3. To connect to the newly created AKS cluster, install the following from your local environment (laptop/pc):
   a. The Kubernetes command-line tool using the instructions provided for your specific OS
   b. The Azure CLI as described in the document, Install the Azure CLI
4. To access the AKS cluster from the terminal, enter `az login` and enter the credentials.
5. Run the following two commands:

   ```
   az account set --subscription xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxxxxxx
   aks get-credentials --resource-group resourcegroup --name aksclustername
   ```
6. Enter Azure CLI: `kubectl get nodes`.

7. If all six nodes are up and running, as shown in the following example, your AKS cluster is ready and connected to your local environment.

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATUS</th>
<th>ROLES</th>
<th>AGE</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>aks-agentpool-34613062-vmss</td>
<td>Ready</td>
<td>agent</td>
<td>22m</td>
<td>v1.18.14</td>
</tr>
<tr>
<td>aks-agentpool-34613062-vmss01</td>
<td>Ready</td>
<td>agent</td>
<td>22m</td>
<td>v1.18.14</td>
</tr>
<tr>
<td>aks-agentpool-34613062-vmss02</td>
<td>Ready</td>
<td>agent</td>
<td>22m</td>
<td>v1.18.14</td>
</tr>
<tr>
<td>aks-gpu-pool-34613062-vmss</td>
<td>Ready</td>
<td>agent</td>
<td>20m</td>
<td>v1.18.14</td>
</tr>
<tr>
<td>aks-gpu-pool-34613062-vmss01</td>
<td>Ready</td>
<td>agent</td>
<td>20m</td>
<td>v1.18.14</td>
</tr>
<tr>
<td>aks-gpu-pool-34613062-vmss02</td>
<td>Ready</td>
<td>agent</td>
<td>20m</td>
<td>v1.18.14</td>
</tr>
</tbody>
</table>

Create a delegated subnet for Azure NetApp Files

To create a delegated subnet for Azure NetApp Files, complete the following steps:

1. Navigate to Virtual Networks within the Azure portal. Find your newly created virtual network. It should have a prefix such as `aks-vnet`.

2. Click the name of the VNet.

3. Click Subnets and click +Subnet from the top toolbar.
4. Provide the subnet with a name such as ANF.sn and, under the Subnet Delegation heading, select Microsoft.Netapp/volumes. Do not change anything else. Click OK.
Azure NetApp Files volumes are allocated to the application cluster and are consumed as persistent volume claims (PVCs) in Kubernetes. In turn, this process provides you the flexibility to map them to different services, such as Jupyter notebooks, serverless functions, and so on.

Users of services can consume storage from the platform in many ways. As this technical report discusses NFSs, the main benefits of Azure NetApp Files are:

- Providing users with the ability to use Snapshot copies.
- Enabling users to store large quantities of data on Azure NetApp Files volumes.
- Using the performance benefits of Azure NetApp Files volumes when running their models on large sets of files.
Peer AKS VNet and Azure NetApp Files VNet

To peer the AKS VNet to the Azure NetApp Files VNet, complete the following steps:

1. Enter Virtual Networks in the search field.
2. Select vnet aks-vnet-name. Click it and enter Peerings in the search field.
3. Click +Add.
4. Enter the following descriptors:
   a. The peering link name is aks-vnet-name_to_anf.
   b. subscriptionID and Azure NetApp Files VNet as the VNet peering partner.
   c. Leave all the nonasterisk sections with the default values.
5. Click Add.

For more information, see Create, change, or delete a virtual network peering.

Install Trident

To install Trident using Helm, complete the following steps:

1. Install Helm (for installation instructions, visit the source).
2. Download and extract the Trident 20.01.1 installer.

   ```bash
   $wget
   $tar -xf trident-installer-21.01.1.tar.gz
   ```

3. Change the directory to trident-installer.

   ```bash
   $cd trident-installer
   ```

4. Copy tridentctl to a directory in your system $PATH.

   ```bash
   $sudo cp ./tridentctl /usr/local/bin
   ```

5. Install Trident on the Kubernetes (K8s) cluster with Helm (source):
   a. Change the directory to the helm directory.

      ```bash
      $cd helm
      ```

   b. Install Trident.
$helm install trident trident-operator-21.01.1.tgz --namespace trident --create-namespace

c. Check the status of Trident pods.

$kubectl -n trident get pods

If all the pods are up and running, then Trident is installed and you can move forward.

6. Set up the Azure NetApp Files backend and storage class for AKS.

   The service principal is how Trident communicates with Azure to manipulate your Azure NetApp Files resources.

   $az ad sp create-for-rbac --name ""

   The output should look like the following example:

   ```
   {
       "appId": "xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx",
       "displayName": "netappprincipal",
       "name": "",
       "password": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
       "tenant": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx"
   }
   ```

7. Create a Trident backend json file, example name anf-backend.json.

8. Using your preferred text editor, complete the following fields inside the anf-backend.json file:
9. Substitute the following fields:
   - subscriptionID. Your Azure subscription ID.
   - tenantID. Your Azure Tenant ID from the output of `az ad sp` in the previous step.
   - clientID. Your appID from the output of `az ad sp` in the previous step.
   - clientSecret. Your password from the output of `az ad sp` in the previous step.

10. Instruct Trident to create the Azure NetApp Files backend in the `trident` namespace using `anf-backend.json` as the configuration file:

    ```bash
    $tridentctl create backend -f anf-backend.json -n trident
    ```

11. Create a storage class. Kubernetes users provision volumes by using PVCs that specify a storage class by name. Instruct K8s to create a storage class `azurenetappfiles` that references the Trident backend created in the previous step.

12. Create a YAML (`anf-storage-class.yaml`) file for storage class and copy.
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: azurenetappfiles
provisioner: netapp.io/trident
parameters:
  backendType: "azure-netapp-files"

$kubectl create -f anf-storage-class.yaml

13. Verify that the storage class was created.

$kubectl get sc azurenetappfiles

<table>
<thead>
<tr>
<th>NAME</th>
<th>PROVISIONER</th>
<th>RECLAIMPOLICY</th>
<th>VOLUMEBINDINGMODE</th>
<th>ALLOWVOLUMEEXPANSION</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>azurenetappfiles</td>
<td>csi.trident.netapp.io</td>
<td>Delete</td>
<td>Immediate</td>
<td>false</td>
<td>98s</td>
</tr>
</tbody>
</table>

Set up Dask with RAPIDS deployment on AKS using Helm

To set up Dask with RAPIDS deployment on AKS using Helm, complete the following steps:

1. Create a namespace for installing Dask with RAPIDS.

$kubectl create namespace rapids-dask

2. Create a PVC to store the click-through rate dataset:

   a. Save the following YAML content to a file to create a PVC.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-criteo-data
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1000Gi
  storageClassName: azurenetappfiles
```

   b. Apply the YAML file to your Kubernetes cluster.
kubectl -n rapids-dask apply -f <your yaml file>


    git clone https://github.com/rapidsai/helm-chart helm-chart

4. Modify values.yaml and include the PVC created earlier for workers and Jupyter workspace.
   a. Go to the rapidsai directory of the repository.

    cd helm-chart/rapidsai

   b. Update the values.yaml file and mount the volume using PVC.

```
dask:
  ...
  worker:
    name: worker
  ...
  mounts:
    volumes:
      - name: data
        persistentVolumeClaim:
          claimName: pvc-criteo-data
        volumeMounts:
          - name: data
            mountPath: /data
  ...

jupyter:
  name: jupyter
  ...
  mounts:
    volumes:
      - name: data
        persistentVolumeClaim:
          claimName: pvc-criteo-data
        volumeMounts:
          - name: data
            mountPath: /data
  ...
```

5. Go to the repository’s home directory and deploy Dask with three worker nodes on AKS using Helm.
cd ..
helm dep update rapidsai
helm install rapids-dask --namespace rapids-dask rapidsai

Azure NetApp Files performance tiers

You can change the service level of an existing volume by moving the volume to another capacity pool that uses the service level you want for the volume. This solution enables customers to start with a small dataset and small number of GPUs in Standard Tier and scale out or scale up to Premium Tier as the amount of data and GPUs increase. The Premium Tier offers four times the throughput per terabyte as the Standard Tier, and scale up is performed without having to move any data to change the service level of a volume.

Dynamically change the service level of a volume

To dynamically change the service level of a volume, complete the following steps:

1. On the Volumes page, right-click the volume whose service level you want to change. Select Change Pool.

2. In the Change Pool window, select the capacity pool to which you want to move the volume.
3. Click OK.

**Automate performance tier change**

The following options are available to automate performance tier changes:

- Dynamic Service Level change is still in Public Preview at this time and not enabled by default. To enable this feature on the Azure Subscription, see this documentation about how to [Dynamically change the service level of a volume](#).

- Azure CLI volume pool change commands are provided in [volume pool change documentation](#) and in the following example:

  ```
  az netappfiles volume pool-change -g mygroup --account-name myaccname --pool-name mypoolname --name myvolname --new-pool-resource-id mynewresourceid
  ```

- PowerShell: The [Set-AzNetAppFilesVolumePool cmdlet](#) changes the pool of an Azure NetApp Files volume and is shown in the following example:
Click through rate prediction data processing and model training

Libraries for data processing and model training

The following table lists the libraries and frameworks that were used to build this task. All these components have been fully integrated with Azure’s role-based access and security controls.

<table>
<thead>
<tr>
<th>Libraries/framework</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dask cuML</td>
<td>For ML to work on GPU, the cuML library provides access to the RAPIDS cuML package with Dask. RAPIDS cuML implements popular ML algorithms, including clustering, dimensionality reduction, and regression approaches, with high-performance GPU-based implementations, offering speed-ups of up to 100x over CPU-based approaches.</td>
</tr>
<tr>
<td>Dask cuDF</td>
<td>cuDF includes various other functions supporting GPU-accelerated extract, transform, load (ETL), such as data subsetting, transformations, one-hot encoding, and more. The RAPIDS team maintains a dask-cudf library that includes helper methods to use Dask and cuDF.</td>
</tr>
<tr>
<td>Scikit Learn</td>
<td>Scikit-learn provides dozens of built-in machine learning algorithms and models, called estimators. Each estimator can be fitted to some data using its <code>fit</code> method.</td>
</tr>
</tbody>
</table>

We used two notebooks to construct the ML pipelines for comparison; one is the conventional Pandas scikit-learn approach, and the other is distributed training with RAPIDS and Dask. Each notebook can be tested individually to see the performance in terms of time and scale. We cover each notebook individually to demonstrate the benefits of distributed training using RAPIDS and Dask.

Load Criteo Click Logs day 15 in Pandas and train a scikit-learn random forest model

This section describes how we used Pandas and Dask DataFrames to load Click Logs data from the Criteo Terabyte dataset. The use case is relevant in digital advertising for ad exchanges to build users’ profiles by predicting whether ads will be clicked or if the exchange isn’t using an accurate model in an automated pipeline.

We loaded day 15 data from the Click Logs dataset, totaling 45GB. Running the following cell in Jupyter
notebook CTR-PandasRF-collated.ipynb creates a Pandas DataFrame that contains the first 50 million rows and generates a scikit-learn random forest model.

```python
%%time
import pandas as pd
import numpy as np
header = ['col'+str(i) for i in range (1,41)] # note that according to criteo, the first column in the dataset is Click Through (CT). Consist of 40 columns
first_row_taken = 50_000_000 # use this in pd.read_csv() if your compute resource is limited.
# total number of rows in day15 is 20B
# take 50M rows

""
Read data & display the following metrics:
1. Total number of rows per day
2. df loading time in the cluster
3. Train a random forest model
""

df = pd.read_csv(file, nrows=first_row_taken, delimiter='\t', names=header)
# take numerical columns
df_sliced = df.iloc[:, 0:14]
# split data into training and Y
Y = df_sliced.pop('col1') # first column is binary (click or not)
# change df_sliced data types & fillna
df_sliced = df_sliced.astype(np.float32).fillna(0)
from sklearn.ensemble import RandomForestClassifier
# Random Forest building parameters
# n_streams = 8 # optimization
max_depth = 10
n_bins = 16
n_trees = 10
rf_model = RandomForestClassifier(max_depth=max_depth, n_estimators=n_trees)
rf_model.fit(df_sliced, Y)
```

To perform prediction by using a trained random forest model, run the following paragraph in this notebook. We took the last one million rows from day 15 as the test set to avoid any duplication. The cell also calculates accuracy of prediction, defined as the percentage of occurrences the model accurately predicts whether a user clicks an ad or not. To review any unfamiliar components in this notebook, see the [official scikit-learn documentation](#).
# testing data, last 1M rows in day15
test_file = '/data/day_15_test'
with open(test_file) as g:
    print(g.readline())

# dataFrame processing for test data
test_df = pd.read_csv(test_file, delimiter='\t', names=header)
test_df_sliced = test_df.iloc[:, 0:14]
test_Y = test_df_sliced.pop('col1')
test_df_sliced = test_df_sliced.astype(np.float32).fillna(0)
# prediction & calculating error
pred_df = rf_model.predict(test_df_sliced)
from sklearn import metrics
# Model Accuracy
print("Accuracy:", metrics.accuracy_score(test_Y, pred_df))

Load Day 15 in Dask and train a Dask cuML random forest model

In a manner similar to the previous section, load Criteo Click Logs day 15 in Pandas and train a scikit-learn random forest model. In this example, we performed DataFrame loading with Dask cuDF and trained a random forest model in Dask cuML. We compared the differences in training time and scale in the section “Training time comparison.”

criteo_dask_RF.ipynb

This notebook imports numpy, cuml, and the necessary dask libraries, as shown in the following example:

```python
import cuml
from dask.distributed import Client, progress, wait
import dask_cudf
import numpy as np
import cudf
from cuml.dask.ensemble import RandomForestClassifier as cumlDaskRF
from cuml.dask.common import import_utils as dask_utils

Initiate Dask Client().

client = Client()
```

If your cluster is configured correctly, you can see the status of worker nodes.
In our AKS cluster, the following status is displayed:

![AKS Cluster Status](image)

Note that Dask employs the lazy execution paradigm: rather than executing the processing code instantly, Dask builds a Directed Acyclic Graph (DAG) of execution instead. DAG contains a set of tasks and their interactions that each worker needs to run. This layout means the tasks do not run until the user tells Dask to execute them in one way or another. With Dask you have three main options:

• **Call compute() on a DataFrame.** This call processes all the partitions and then returns results to the scheduler for final aggregation and conversion to cuDF DataFrame. This option should be used sparingly and only on heavily reduced results unless your scheduler node runs out of memory.

• **Call persist() on a DataFrame.** This call executes the graph, but, instead of returning the results to the scheduler node, it maintains them across the cluster in memory so the user can reuse these intermediate results down the pipeline without the need for rerunning the same processing.

• **Call head() on a DataFrame.** Just like with cuDF, this call returns 10 records back to the scheduler node. This option can be used to quickly check if your DataFrame contains the desired output format, or if the records themselves make sense, depending on your processing and calculation.

Therefore, unless the user calls either of these actions, the workers sit idle waiting for the scheduler to initiate the processing. This lazy execution paradigm is common in modern parallel and distributed computing frameworks such as Apache Spark.

The following paragraph trains a random forest model by using Dask cuML for distributed GPU-accelerated computing and calculates model prediction accuracy.
Monitor Dask using native Task Streams dashboard

The Dask distributed scheduler provides live feedback in two forms:

- An interactive dashboard containing many plots and tables with live information
- A progress bar suitable for interactive use in consoles or notebooks

In our case, the following figure shows how you can monitor the task progress, including Bytes Stored, the Task Stream with a detailed breakdown of the number of streams, and Progress by task names with associated functions executed. In our case, because we have three worker nodes, there are three main chunks of stream and the color codes denote different tasks within each stream.

You have the option to analyze individual tasks and examine the execution time in milliseconds or identify any obstacles or hindrances. For example, the following figure shows the Task Streams for the random forest model fitting stage. There are considerably more functions being executed, including unique chunk for DataFrame processing, _construct_rf for fitting the random forest, and so on. Most of the time was spent on DataFrame operations due to the large size (45GB) of one day of data from the Criteo Click Logs.
Training time comparison

This section compares the model training time using conventional Pandas compared to Dask. For Pandas, we loaded a smaller amount of data due to the nature of slower processing time to avoid memory overflow. Therefore, we interpolated the results to offer a fair comparison.

The following table shows the raw training time comparison when there is significantly less data used for the Pandas random forest model (50 million rows out of 20 billion per day15 of the dataset). This sample is only using less than 0.25% of all available data. Whereas for Dask-cuML we trained the random forest model on all 20 billion available rows. The two approaches yielded comparable training time.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Training time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scikit-learn: Using only 50M rows in day15 as the training data</td>
<td>47 minutes and 21 seconds</td>
</tr>
<tr>
<td>RAPIDS-Dask: Using all 20B rows in day15 as the training data</td>
<td>1 hour, 12 minutes, and 11 seconds</td>
</tr>
</tbody>
</table>

If we interpolate the training time results linearly, as shown in the following table, there is a significant advantage to using distributed training with Dask. It would take the conventional Pandas scikit-learn approach 13 days to process and train 45GB of data for a single day of click logs, whereas the RAPIDS-Dask approach processes the same amount of data 262.39 times faster.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Training time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scikit-learn: Using all 20B rows in day15 as the training data</td>
<td>13 days, 3 hours, 40 minutes, and 11 seconds</td>
</tr>
<tr>
<td>RAPIDS-Dask: Using all 20B rows in day15 as the training data</td>
<td>1 hour, 12 minutes, and 11 seconds</td>
</tr>
</tbody>
</table>

In the previous table, you can see that by using RAPIDS with Dask to distribute the data processing and model training across multiple GPU instances, the run time is significantly shorter compared to conventional Pandas DataFrame processing with scikit-learn model training. This framework enables scaling up and out in the cloud.
as well as on-premises in a multinode, multi-GPU cluster.

Monitor Dask and RAPIDS with Prometheus and Grafana

After everything is deployed, run inferences on new data. The models predict whether a user clicks an ad based on browsing activities. The results of the prediction are stored in a Dask cuDF. You can monitor the results with Prometheus and visualize in Grafana dashboards.

For more information, see this RAPIDS AI Medium post.

Dataset and model versioning using NetApp DataOps Toolkit

The NetApp DataOps Toolkit for Kubernetes abstracts storage resources and Kubernetes workloads up to the data-science workspace level. These capabilities are packaged in a simple, easy-to-use interface that is designed for data scientists and data engineers. Using the familiar form of a Python program, the Toolkit enables data scientists and engineers to provision and destroy JupyterLab workspaces in just seconds. These workspaces can contain terabytes, or even petabytes, of storage capacity, enabling data scientists to store all their training datasets directly in their project workspaces. Gone are the days of separately managing workspaces and data volumes.

For more information, visit the Toolkit’s GitHub repository.

Jupyter notebooks for reference

There are two Jupyter notebooks associated with this technical report:

• CTR-PandasRF-collated.ipynb. This notebook loads Day 15 from the Criteo Terabyte Click Logs dataset, processes and formats data into a Pandas DataFrame, trains a Scikit-learn random forest model, performs prediction, and calculates accuracy.

• criteo_dask_RF.ipynb. This notebook loads Day 15 from the Criteo Terabyte Click Logs dataset, processes and formats data into a Dask cuDF, trains a Dask cuML random forest model, performs prediction, and calculates accuracy. By leveraging multiple worker nodes with GPUs, this distributed data and model processing and training approach is highly efficient. The more data you process, the greater the time savings versus a conventional ML approach. You can deploy this notebook in the cloud, on-premises, or in a hybrid environment where your Kubernetes cluster contains compute and storage in different locations, as long as your networking setup enables the free movement of data and model distribution.

Conclusion

Azure NetApp Files, RAPIDS, and Dask speed up and simplify the deployment of large-scale ML processing and training by integrating with orchestration tools such as Docker and Kubernetes. By unifying the end-to-end data pipeline, this solution reduces the latency and complexity inherent in many advanced computing workloads, effectively bridging the gap between development and operations. Data scientists can run queries on large datasets and securely share data and algorithmic models with other users during the training phase.

When building your own AI/ML pipelines, configuring the integration, management, security, and accessibility of
the components in an architecture is a challenging task. Giving developers access and control of their environment presents another set of challenges.

By building an end-to-end distributed training model and data pipeline in the cloud, we demonstrated two orders of magnitude improvement in total workflow completion time versus a conventional, open-source approach that did not leverage GPU-accelerated data processing and compute frameworks.

The combination of NetApp, Microsoft, opens-source orchestration frameworks, and NVIDIA brings the latest technologies together as managed services with great flexibility to accelerate technology adoption and improve the time to market for new AI/ML applications. These advanced services are delivered in a cloud-native environment that can be easily ported for on-premises as well as hybrid deployment architectures.

Where to find additional information

To learn more about the information that is described in this document, see the following resources:

• Azure NetApp Files:
  ◦ Solutions architecture page for Azure NetApp Files
    https://docs.microsoft.com/azure/azure-netapp-files/azure-netapp-files-solution-architectures

• Trident persistent storage for containers:
  ◦ Azure NetApp Files and Trident

• Dask and RAPIDS:
  ◦ Dask
    https://docs.dask.org/en/latest/
  ◦ Install Dask
    https://docs.dask.org/en/latest/install.html
  ◦ Dask API
    https://docs.dask.org/en/latest/api.html
  ◦ Dask Machine Learning
    https://examples.dask.org/machine-learning.html
  ◦ Dask Distributed Diagnostics

• ML framework and tools:
    https://www.tensorflow.org/
TR-4896: Distributed training in Azure: Lane detection - Solution design

Muneer Ahmad and Verron Martina, NetApp
Ronen Dar, RUN:AI

Since May 2019, Microsoft delivers an Azure native, first-party portal service for enterprise NFS and SMB file services based on NetApp ONTAP technology. This development is driven by a strategic partnership between Microsoft and NetApp and further extends the reach of world-class ONTAP data services to Azure.

NetApp, a leading cloud data services provider, has teamed up with RUN: AI, a company virtualizing AI infrastructure, to allow faster AI experimentation with full GPU utilization. The partnership enables teams to speed up AI by running many experiments in parallel, with fast access to data, and leveraging limitless compute resources. RUN: AI enables full GPU utilization by automating resource allocation, and the proven architecture of Azure NetApp Files enables every experiment to run at maximum speed by eliminating data pipeline obstructions.

NetApp and RUN: AI have joined forces to offer customers a future-proof platform for their AI journey in Azure. From analytics and high-performance computing (HPC) to autonomous decisions (where customers can optimize their IT investments by only paying for what they need, when they need it), the alliance between NetApp and RUN: AI offers a single unified experience in the Azure Cloud.

Solution overview

In this architecture, the focus is on the most computationally intensive part of the AI or machine learning (ML) distributed training process of lane detection. Lane detection is one of the most important tasks in autonomous driving, which helps to guide vehicles by localization of the lane markings. Static components like lane markings guide the vehicle to drive on the highway interactively and safely.

Convolutional Neural Network (CNN)-based approaches have pushed scene understanding and segmentation to a new level. Although it doesn’t perform well for objects with long structures and regions that could be occluded (for example, poles, shade on the lane, and so on). Spatial Convolutional Neural Network (SCNN) generalizes the CNN to a rich spatial level. It allows information propagation between neurons in the same layer, which makes it best suited for structured objects such as lanes, poles, or truck with occlusions. This compatibility is because the spatial information can be reinforced, and it preserves smoothness and continuity.
Thousands of scene images need to be injected in the system to allow the model learn and distinguish the various components in the dataset. These images include weather, daytime or nighttime, multilane highway roads, and other traffic conditions.

For training, there is a need for good quality and quantity of data. Single GPU or multiple GPUs can take days to weeks to complete the training. Data-distributed training can speed up the process by using multiple and multinode GPUs. Horovod is one such framework that grants distributed training but reading data across clusters of GPUs could act as a hindrance. Azure NetApp Files provides ultrafast, high throughput and sustained low latency to provide scale-out/scale-up capabilities so that GPUs are leveraged to the best of their computational capacity. Our experiments verified that all the GPUs across the cluster are used more than 96% on average for training the lane detection using SCNN.

Target audience

Data science incorporates multiple disciplines in IT and business, therefore multiple personas are part of our targeted audience:

- Data scientists need the flexibility to use the tools and libraries of their choice.
- Data engineers need to know how the data flows and where it resides.
- Autonomous driving use-case experts.
- Cloud administrators and architects to set up and manage cloud (Azure) resources.
- A DevOps engineer needs the tools to integrate new AI/ML applications into their continuous integration and continuous deployment (CI/CD) pipelines.
- Business users want to have access to AI/ML applications.

In this document, we describe how Azure NetApp Files, RUN: AI, and Microsoft Azure help each of these roles bring value to business.

Solution technology

This section covers the technology requirements for the lane detection use case by implementing a distributed training solution at scale that fully runs in the Azure cloud. The figure below provides an overview of the solution architecture.

The elements used in this solution are:

- Azure Kubernetes Service (AKS)
- Azure Compute SKUs with NVIDIA GPUs
- Azure NetApp Files
- RUN: AI
- NetApp Trident

Links to all the elements mentioned here are listed in the Additional information section.
Cloud resources and services requirements

The following table lists the hardware components that are required to implement the solution. The cloud components that are used in any implementation of the solution might vary based on customer requirements.

<table>
<thead>
<tr>
<th>Cloud</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>AKS</td>
<td>Minimum of three system nodes and three GPU worker nodes</td>
</tr>
<tr>
<td>Virtual machine (VM) SKU system nodes</td>
<td>Three Standard_DS2_v2</td>
</tr>
<tr>
<td>VM SKU GPU worker nodes</td>
<td>Three Standard_NC6s_v3</td>
</tr>
<tr>
<td>Azure NetApp Files</td>
<td>4TB standard tier</td>
</tr>
</tbody>
</table>

Software requirements

The following table lists the software components that are required to implement the solution. The software components that are used in any implementation of the solution might vary based on customer requirements.

<table>
<thead>
<tr>
<th>Software</th>
<th>Version or other information</th>
</tr>
</thead>
<tbody>
<tr>
<td>AKS - Kubernetes version</td>
<td>1.18.14</td>
</tr>
<tr>
<td>RUN:AI CLI</td>
<td>v2.2.25</td>
</tr>
<tr>
<td>RUN:AI Orchestration Kubernetes Operator version</td>
<td>1.0.109</td>
</tr>
<tr>
<td>Horovod</td>
<td>0.21.2</td>
</tr>
<tr>
<td>NetApp Trident</td>
<td>20.01.1</td>
</tr>
<tr>
<td>Helm</td>
<td>3.0.0</td>
</tr>
</tbody>
</table>

Lane detection – Distributed training with RUN:AI

This section provides details on setting up the platform for performing lane detection distributed training at scale using the RUN: AI orchestrator. We discuss installation of all the solution elements and running the distributed training job on the said platform. ML
versioning is completed by using NetApp SnapshotTM linked with RUN: AI experiments for achieving data and model reproducibility. ML versioning plays a crucial role in tracking models, sharing work between team members, reproducibility of results, rolling new model versions to production, and data provenance. NetApp ML version control (Snapshot) can capture point-in-time versions of the data, trained models, and logs associated with each experiment. It has rich API support making it easy to integrate with the RUN: AI platform; you just have to trigger an event based on the training state. You also have to capture the state of the whole experiment without changing anything in the code or the containers running on top of Kubernetes (K8s).

Finally, this technical report wraps up with performance evaluation on multiple GPU-enabled nodes across AKS.

**Distributed training for lane detection use case using the TuSimple dataset**

In this technical report, distributed training is performed on the TuSimple dataset for lane detection. Horovod is used in the training code for conducting data distributed training on multiple GPU nodes simultaneously in the Kubernetes cluster through AKS. Code is packaged as container images for TuSimple data download and processing. Processed data is stored on persistent volumes allocated by NetApp Trident plug-in. For the training, one more container image is created, and it uses the data stored on persistent volumes created during downloading the data.

To submit the data and training job, use RUN: AI for orchestrating the resource allocation and management. RUN: AI allows you to perform Message Passing Interface (MPI) operations which are needed for Horovod. This layout allows multiple GPU nodes to communicate with each other for updating the training weights after every training mini batch. It also enables monitoring of training through the UI and CLI, making it easy to monitor the progress of experiments.

NetApp Snapshot is integrated within the training code and captures the state of data and the trained model for every experiment. This capability enables you to track the version of data and code used, and the associated trained model generated.

**AKS setup and installation**

For setup and installation of the AKS cluster go to Create an AKS Cluster. Then, follow these series of steps:

1. When selecting the type of nodes (whether it be system (CPU) or worker (GPU) nodes), select the following:
   a. Add primary system node named agentpool at the Standard_DS2_v2 size. Use the default three nodes.
   b. Add worker node gpupool with the Standard_NC6s_v3 pool size. Use three nodes minimum for GPU nodes.
Deployment takes 5–10 minutes.

2. After deployment is complete, click Connect to Cluster. To connect to the newly created AKS cluster, install the Kubernetes command-line tool from your local environment (laptop/PC). Visit Install Tools to install it as per your OS.

3. Install Azure CLI on your local environment.

4. To access the AKS cluster from the terminal, first enter `az login` and put in the credentials.

5. Run the following two commands:

   ```
   az account set --subscription xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx
   aks get-credentials --resource-group resourcegroup --name aksclusternane
   ```

6. Enter this command in the Azure CLI:

   ```
   kubectl get nodes
   ```

   If all six nodes are up and running as seen here, your AKS cluster is ready and connected to your local environment.

---

Create a delegated subnet for Azure NetApp Files

To create a delegated subnet for Azure NetApp Files, follow this series of steps:

1. Navigate to Virtual networks within the Azure portal. Find your newly created virtual network. It should have a prefix such as aks-vnet, as seen here. Click the name of the virtual network.
2. Click Subnets and select +Subnet from the top toolbar.

3. Provide the subnet with a name such as ANF.sn and under the Subnet Delegation heading, select Microsoft.NetApp/volumes. Do not change anything else. Click OK.
Azure NetApp Files volumes are allocated to the application cluster and are consumed as persistent volume claims (PVCs) in Kubernetes. In turn, this allocation provides us the flexibility to map volumes to different services, be it Jupyter notebooks, serverless functions, and so on.

Users of services can consume storage from the platform in many ways. The main benefits of Azure NetApp Files are:

- Provides users with the ability to use snapshots.
- Enables users to store large quantities of data on Azure NetApp Files volumes.
- Procure the performance benefits of Azure NetApp Files volumes when running their models on large sets of files.
Azure NetApp Files setup

To complete the setup of Azure NetApp Files, you must first configure it as described in Quickstart: Set up Azure NetApp Files and create an NFS volume.

However, you may omit the steps to create an NFS volume for Azure NetApp Files as you will create volumes through Trident. Before continuing, be sure that you have:

2. Created an account in Azure NetApp Files.
3. Set up a capacity pool (minimum 4TiB Standard or Premium depending on your needs).

Peering of AKS virtual network and Azure NetApp Files virtual network

Next, peer the AKS virtual network (VNet) with the Azure NetApp Files VNet by following these steps:

1. In the search box at the top of the Azure portal, type virtual networks.
2. Click VNet aks- vnet-name, then enter Peerings in the search field.
3. Click +Add and enter the information provided in the table below:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value or description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peering link name</td>
<td>aks-vnet-name_to_anf</td>
</tr>
<tr>
<td>SubscriptionID</td>
<td>Subscription of the Azure NetApp Files VNet to which you’re peering</td>
</tr>
<tr>
<td>VNet peering partner</td>
<td>Azure NetApp Files VNet</td>
</tr>
</tbody>
</table>

Leave all the nonasterisk sections on default

4. Click ADD or OK to add the peering to the virtual network.

For more information, visit Create, change, or delete a virtual network peering.

Trident

Trident is an open-source project that NetApp maintains for application container persistent storage. Trident has been implemented as an external provisioner controller that runs as a pod itself, monitoring volumes and completely automating the provisioning process.

NetApp Trident enables smooth integration with K8s by creating and attaching persistent volumes for storing training datasets and trained models. This capability makes it easier for data scientists and data engineers to use K8s without the hassle of manually storing and managing datasets. Trident also eliminates the need for data scientists to learn managing new data platforms as it integrates the data management-related tasks through the logical API integration.

Install Trident

To install Trident software, complete the following steps:

1. First install helm.
2. Download and extract the Trident 21.01.1 installer.
tar -xf trident-installer-21.01.1.tar.gz

3. Change the directory to trident-installer.

   cd trident-installer

4. Copy tridentctl to a directory in your system $PATH.

   cp ./tridentctl /usr/local/bin

5. Install Trident on K8s cluster with Helm:
   a. Change directory to helm directory.

      cd helm

   b. Install Trident.

      helm install trident trident-operator-21.01.1.tgz --namespace trident
          --create-namespace

   c. Check the status of Trident pods the usual K8s way:

      kubectl -n trident get pods

   d. If all the pods are up and running, Trident is installed and you are good to move forward.

Set up Azure NetApp Files back-end and storage class

To set up Azure NetApp Files back-end and storage class, complete the following steps:

1. Switch back to the home directory.

   cd ~

2. Clone the project repository lane-detection-SCNN-horovod.

3. Go to the trident-config directory.
cd ./lane-detection-SCNN-horovod/trident-config

4. Create an Azure Service Principle (the service principle is how Trident communicates with Azure to access your Azure NetApp Files resources).

   az ad sp create-for-rbac --name

   The output should look like the following example:

   ```
   {
     "appId": "xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
     "displayName": "netapptrident",
     "name": "http://netapptrident",
     "password": "xxxxxxxxxxxxxxxx.xxxxxxxxxxxxxx",
     "tenant": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx"
   }
   ```

5. Create the Trident backend json file.

6. Using your preferred text editor, complete the following fields from the table below inside the anf-backend.json file.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>subscriptionID</td>
<td>Your Azure Subscription ID</td>
</tr>
<tr>
<td>tenantID</td>
<td>Your Azure Tenant ID (from the output of az ad sp in the previous step)</td>
</tr>
<tr>
<td>clientId</td>
<td>Your appID (from the output of az ad sp in the previous step)</td>
</tr>
<tr>
<td>clientSecret</td>
<td>Your password (from the output of az ad sp in the previous step)</td>
</tr>
</tbody>
</table>

   The file should look like the following example:
7. Instruct Trident to create the Azure NetApp Files back-end in the `trident` namespace, using `anf-backend.json` as the configuration file as follows:

```
tridentctl create backend -f anf-backend.json -n trident
```

8. Create the storage class:
   a. K8s users provision volumes by using PVCs that specify a storage class by name. Instruct K8s to create a storage class `azurenetappfiles` that will reference the Azure NetApp Files back-end created in the previous step using the following:

```
kubectl create -f anf-storage-class.yaml
```

   b. Check that storage class is created by using the following command:

```
kubectl get sc azurenetappfiles
```

The output should look like the following example:

<table>
<thead>
<tr>
<th>Name</th>
<th>Provisioner</th>
<th>Reclaim Policy</th>
<th>Volume Binding Mode</th>
<th>Allow Volume Expansion</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>azurenetappfiles</td>
<td>csi.trident.netapp.io</td>
<td>Delete</td>
<td>Immediate</td>
<td>false</td>
<td>98s</td>
</tr>
</tbody>
</table>

**Deploy and set up volume snapshot components on AKS**

If your cluster does not come pre-installed with the correct volume snapshot components, you may manually install these components by running the following steps:
AKS 1.18.14 does not have pre-installed Snapshot Controller.

1. Install Snapshot Beta CRDs by using the following commands:

   ```shell
   kubectl create -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotclasses.yaml
   kubectl create -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotcontents.yaml
   kubectl create -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshots.yaml
   ```

2. Install Snapshot Controller by using the following documents from GitHub:

   ```shell
   ```

3. Set up K8s volumesnapshotclass: Before creating a volume snapshot, a volume snapshot class must be set up. Create a volume snapshot class for Azure NetApp Files, and use it to achieve ML versioning by using NetApp Snapshot technology. Create volumesnapshotclass netapp-csi-snapclass and set it to default `volumesnapshotclass` as such:

   ```shell
   kubectl create -f netapp-volume-snapshot-class.yaml
   ```

   The output should look like the following example:

   ```
   volumesnapshotclass.snapshot.storage.k8s.io/netapp-csi-snapclass created
   ```

4. Check that the volume Snapshot copy class was created by using the following command:

   ```shell
   kubectl get volumesnapshotclass
   ```

   The output should look like the following example:

   ```
<table>
<thead>
<tr>
<th>NAME</th>
<th>DRIVER</th>
<th>DELETIONPOLICY</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>netapp-csi-snapclass</td>
<td>csi.trident.netapp.io</td>
<td>Delete</td>
<td>63s</td>
</tr>
</tbody>
</table>
   ```
RUN:AI installation

To install RUN:AI, complete the following steps:

1. Install RUN:AI cluster on AKS.

2. Go to app.runai.ai, click create New Project, and name it lane-detection. It will create a namespace on a K8s cluster starting with runai- followed by the project name. In this case, the namespace created would be runai-lane-detection.

3. Install RUN:AI CLI.

4. On your terminal, set lane-detection as a default RUN: AI project by using the following command:

   ```
   `runai config project lane-detection`
   ```

   The output should look like the following example:

   **Project lane-detection has been set as default project**
5. Create ClusterRole and ClusterRoleBinding for the project namespace (for example, lane-detection) so the default service account belonging to runai-lane-detection namespace has permission to perform volumesnapshot operations during job execution:

   a. List namespaces to check that runai-lane-detection exists by using this command:

   ```
kubectl get namespaces
   ```

   The output should appear like the following example:

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATUS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>Active</td>
<td>130m</td>
</tr>
<tr>
<td>kube-node-lease</td>
<td>Active</td>
<td>130m</td>
</tr>
<tr>
<td>kube-public</td>
<td>Active</td>
<td>130m</td>
</tr>
<tr>
<td>kube-system</td>
<td>Active</td>
<td>130m</td>
</tr>
<tr>
<td>runai</td>
<td>Active</td>
<td>4m44s</td>
</tr>
<tr>
<td>runai-lane-detection</td>
<td>Active</td>
<td>13s</td>
</tr>
<tr>
<td>trident</td>
<td>Active</td>
<td>102m</td>
</tr>
</tbody>
</table>

6. Create ClusterRole netappsnapshot and ClusterRoleBinding netappsnapshot using the following commands:

   ```
kubectl create -f runai-project-snap-role.yaml
kubectl create -f runai-project-snap-role-binding.yaml
   ```

Download and process the TuSimple dataset as RUN:AI job

The process to download and process the TuSimple dataset as a RUN:AI job is optional. It involves the following steps:

1. Build and push the docker image, or omit this step if you want to use an existing docker image (for example, muneer7589/download-tusimple:1.0)
   a. Switch to the home directory:

   ```
cd ~
   ```

   b. Go to the data directory of the project lane-detection-SCNN-horovod:

   ```
cd ./lane-detection-SCNN-horovod/data
   ```

   c. Modify build_image.sh shell script and change docker repository to yours. For example, replace muneer7589 with your docker repository name. You could also change the docker image name and
d. Run the script to build the docker image and push it to the docker repository using these commands:

```bash
chmod +x build_image.sh
./build_image.sh
```

2. Submit the RUN: AI job to download, extract, pre-process, and store the TuSimple lane detection dataset in a pvc, which is dynamically created by NetApp Trident:
   a. Use the following commands to submit the RUN: AI job:

```bash
runai submit
--name download-tusimple-data
--pvc azurenetappfiles:100Gi:/mnt
--image muneer7589/download-tusimple:1.0
```

b. Enter the information from the table below to submit the RUN:AI job:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value or description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-name</td>
<td>Name of the job</td>
</tr>
<tr>
<td>Field</td>
<td>Value or description</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------</td>
</tr>
<tr>
<td>-pvc</td>
<td>PVC of the format [StorageClassName]:Size:ContainerMountPath</td>
</tr>
<tr>
<td></td>
<td>In the above job submission, you are creating an PVC based on-demand using Trident with storage class azurenetappfiles. Persistent volume capacity here is 100Gi and it's mounted at path /mnt.</td>
</tr>
<tr>
<td>-image</td>
<td>Docker image to use when creating the container for this job</td>
</tr>
</tbody>
</table>

The output should look like the following example:

```
The job 'download-tusimple-data' has been submitted successfully
You can run `runai describe job download-tusimple-data -p lane-detection` to check the job status
```

c. List the submitted RUN:AI jobs.

```
runai list jobs
```

d. Check the submitted job logs.

```
runai logs download-tusimple-data -t 10
```

e. List the pvc created. Use this pvc command for training in the next step.

```
kubectl get pvc | grep download-tusimple-data
```

The output should look like the following example:

```
pvc-download-tusimple-data-0 Bound  pvc-6b06374d-2c17-40c8-a445-79f3b5816d50  100Gi  RWO  azurenetappfiles 4m47s
```

f. Check the job in RUN: AI UI (or app.run.ai).
Perform distributed lane detection training using Horovod

Performing distributed lane detection training using Horovod is an optional process. However, here are the steps involved:

1. Build and push the docker image, or skip this step if you want to use the existing docker image (for example, muneer7589/dist-lane-detection:3.1):
   a. Switch to home directory.

   ```sh
cd ~
```

   b. Go to the project directory `lane-detection-SCNN-horovod`.

   ```sh
cd ./lane-detection-SCNN-horovod
```

   c. Modify the `build_image.sh` shell script and change docker repository to yours (for example, replace muneer7589 with your docker repository name). You could also change the docker image name and TAG (dist-lane-detection and 3.1, for example).
d. Run the script to build the docker image and push to the docker repository.

```
chmod +x build_image.sh
./build_image.sh
```

2. Submit the RUN: AI job for carrying out distributed training (MPI):

a. Using submit of RUN: AI for automatically creating PVC in the previous step (for downloading data) only allows you to have RWO access, which does not allow multiple pods or nodes to access the same PVC for distributed training. Update the access mode to ReadWriteMany and use the Kubernetes patch to do so.

b. First, get the volume name of the PVC by running the following command:

```
kubectl get pvc | grep download-tusimple-data
```

c. Patch the volume and update access mode to ReadWriteMany (replace volume name with yours in the following command):

```
kubectl patch pv pvc-bb03b74d-2c17-40c4-a445-79f3de8d16d5 -p '"
.spec":{"accessModes":["ReadWriteMany"]}"
```

d. Submit the RUN: AI MPI job for executing the distributed training job using information from the table below:
runai submit-mpi
--name dist-lane-detection-training
--large-shm
--processes=3
--gpu 1
--pvc pvc-download-tusimple-data-0:/mnt
--image muneer7589/dist-lane-detection:3.1
-e USE_WORKERS="true"
-e NUM_WORKERS=4
-e BATCH_SIZE=33
-e USE_VAL="false"
-e VAL_BATCH_SIZE=99
-e ENABLE_SNAPSHOT="true"
-e PVC_NAME="pvc-download-tusimple-data-0"

<table>
<thead>
<tr>
<th>Field</th>
<th>Value or description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the distributed training job</td>
</tr>
<tr>
<td>large shm</td>
<td>Mount a large /dev/shm device</td>
</tr>
<tr>
<td></td>
<td>It is a shared file system mounted on RAM and provides large enough shared memory for multiple CPU workers to process and load batches into CPU RAM.</td>
</tr>
<tr>
<td>processes</td>
<td>Number of distributed training processes</td>
</tr>
<tr>
<td>gpu</td>
<td>Number of GPUs/processes to allocate for the job</td>
</tr>
<tr>
<td></td>
<td>In this job, there are three GPU worker processes (--processes=3), each allocated with a single GPU (--gpu 1)</td>
</tr>
<tr>
<td>pvc</td>
<td>Use existing persistent volume (pvc-download-tusimple-data-0) created by previous job (download-tusimple-data) and it is mounted at path /mnt</td>
</tr>
<tr>
<td>image</td>
<td>Docker image to use when creating the container for this job</td>
</tr>
</tbody>
</table>

 Define environment variables to be set in the container

- **USE_WORKERS**
  - Setting the argument to true turns on multi-process data loading

- **NUM_WORKERS**
  - Number of data loader worker processes

- **BATCH_SIZE**
  - Training batch size

- **USE_VAL**
  - Setting the argument to true allows validation

- **VAL_BATCH_SIZE**
  - Validation batch size
### Field | Value or description
--- | ---
ENABLE_SNAPSHOT | Setting the argument to true enables taking data and trained model snapshots for ML versioning purposes
PVC_NAME | Name of the pvc to take a snapshot of. In the above job submission, you are taking a snapshot of pvc-download-tusimple-data-0, consisting of dataset and trained models

The output should look like the following example:

**The job 'dist-lane-detection-training' has been submitted successfully**
*You can run `runai describe job dist-lane-detection-training --p lane-detection` to check the job status*

e. List the submitted job.

    runai list jobs

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATUS</th>
<th>AGE</th>
<th>IMAGE</th>
<th>TYPE</th>
<th>PROJECT</th>
<th>USER</th>
<th>GPUs Allocated (Requested)</th>
<th>PODs</th>
</tr>
</thead>
<tbody>
<tr>
<td>download-tusimple-data</td>
<td>Succeeded</td>
<td>Id</td>
<td>munge7509/download-tusimple:1.0</td>
<td>Train</td>
<td>lane-detection</td>
<td>verronmartina</td>
<td>0 (0)</td>
<td>0</td>
</tr>
<tr>
<td>dist-lane-detection-training</td>
<td>Initiated</td>
<td>2m</td>
<td>munge7509/dist-lane-detection:3.1</td>
<td>Train</td>
<td>lane-detection</td>
<td>root</td>
<td>3 (3)</td>
<td>4 (0)</td>
</tr>
</tbody>
</table>

f. Submitted job logs:

    runai logs dist-lane-detection-training

```
runai logs dist-lane-detection-training
Running with 3 workers
2021-03-04 17:29:23.158449: I tensorflow/stream_executor/platform/default/dso_loader.cc:43] Successfully opened dynamic library libcuda.so.10.1
  [ d = - ]
  shift
  /opt/kube/kubectl cp /opt/kube/hosts dist-lane-detection-training-worker-0:/etc/hosts_of_nodes
  POD_NAME=dist-lane-detection-training-worker-2
  [ d = - ]
  shift
  /opt/kube/kubectl cp /opt/kube/hosts dist-lane-detection-training-worker-2:/etc/hosts_of_nodes
  POD_NAME=dist-lane-detection-training-worker-3
```

g. Check training job in RUN: AI GUI (or app.runai.ai): RUN: AI Dashboard, as seen in the figures below. The first figure details three GPUs allocated for the distributed training job spread across three nodes on AKS, and the second RUN:AI jobs:
h. After the training is finished, check the NetApp Snapshot copy that was created and linked with RUN: AI job.

```bash
runai logs dist-lane-detection-training --tail 1
```

```
kubectl get volumesnapshots | grep download-tusimple-data-0
```

```
kubectl get volumesnapshots | grep download-tusimple-data-0
```
Restore data from the NetApp Snapshot copy

To restore data from the NetApp Snapshot copy, complete the following steps:

1. Switch to home directory.

   
   cd ~

2. Go to the project directory `lane-detection-SCNN-horovod`.

   
   cd ./lane-detection-SCNN-horovod

3. Modify `restore-snapshot-pvc.yaml` and update `dataSource name` field to the Snapshot copy from which you want to restore data. You could also change PVC name where the data will be restored to, in this example its `restored-tusimple`.

   ```yaml
   apiVersion: v1
   kind: PersistentVolumeClaim
   metadata:
     name: restored-tusimple
   spec:
     storageClassName: azurenetappfiles
     dataSource:
       kind: VolumeSnapshot
     apiGroup: snapshot.storage.k8s.io
     accessModes:
       - ReadWriteMany
     resources:
       requests:
         storage: 100Gi
   ```

4. Create a new PVC by using `restore-snapshot-pvc.yaml`.

   
   kubectl create -f restore-snapshot-pvc.yaml

   The output should look like the following example:

   ```
persistentvolumeclaim/restored-tusimple created
```

5. If you want to use the just restored data for training, job submission remains the same as before; only replace the `PVC_NAME` with the restored `PVC_NAME` when submitting the training job, as seen in the following commands:
Performance evaluation

To show the linear scalability of the solution, performance tests have been done for two scenarios: one GPU and three GPUs. GPU allocation, GPU and memory utilization, different single- and three-node metrics have been captured during the training on the TuSimple lane detection dataset. Data is increased five-fold just for the sake of analyzing resource utilization during the training processes.

The solution enables customers to start with a small dataset and a few GPUs. When the amount of data and the demand of GPUs increase, customers can dynamically scale out the terabytes in the Standard Tier and quickly scale up to the Premium Tier to get four times the throughput per terabyte without moving any data. This process is further explained in the section, Azure NetApp Files service levels.

Processing time on one GPU was 12 hours and 45 minutes. Processing time on three GPUs across three nodes was approximately 4 hours and 30 minutes.

The figures shown throughout the remainder of this document illustrate examples of performance and scalability based on individual business needs.

The figure below illustrates 1 GPU allocation and memory utilization.
The figure below illustrates single node GPU utilization.

The figure below illustrates single node memory size (16GB).
The figure below illustrates single node GPU count (1).

The figure below illustrates single node GPU allocation (%).

The figure below illustrates three GPUs across three nodes – GPUs allocation and memory.
The figure below illustrates three GPUs across three nodes utilization (%).

The figure below illustrates three GPUs across three nodes memory utilization (%).
Azure NetApp Files service levels

You can change the service level of an existing volume by moving the volume to another capacity pool that uses the service level you want for the volume. This existing service-level change for the volume does not require that you migrate data. It also does not affect access to the volume.

Dynamically change the service level of a volume

To change the service level of a volume, use the following steps:

1. On the Volumes page, right-click the volume whose service level you want to change. Select Change Pool.

2. In the Change Pool window, select the capacity pool you want to move the volume to. Then, click OK.
Automate service level change

Dynamic Service Level change is currently still in Public Preview, but it is not enabled by default. To enable this feature on the Azure subscription, follow these steps provided in the document “Dynamically change the service level of a volume.”

• You can also use the following commands for Azure: CLI. For more information about changing the pool size of Azure NetApp Files, visit az netappfiles volume: Manage Azure NetApp Files (ANF) volume resources.

   az netappfiles volume pool-change -g mygroup
   --account-name myaccname
   -pool-name mypoolname
   --name myvolname
   --new-pool-resource-id mynewresourceid

• The set- aznetappfilesvolumepool cmdlet shown here can change the pool of an Azure NetApp Files volume. More information about changing volume pool size and Azure PowerShell can be found by visiting Change pool for an Azure NetApp Files volume.
Set-AzNetAppFilesVolumePool
-ResourceGroupName "MyRG"
-AccountName "MyAnfAccount"
-PoolName "MyAnfPool"
-Name "MyAnfVolume"
-NewPoolResourceId 7d6e4069-6c78-6c61-7bf6-c60968e45fbf

Conclusion

NetApp and RUN: AI have partnered in the creation of this technical report to demonstrate the unique capabilities of the Azure NetApp Files together with the RUN: AI platform for simplifying orchestration of AI workloads. This technical report provides a reference architecture for streamlining the process of both data pipelines and workload orchestration for distributed lane detection training.

In conclusion, with regard to distributed training at scale (especially in a public cloud environment), the resource orchestration and storage component is a critical part of the solution. Making sure that data managing never hinders multiple GPU processing, therefore results in the optimal utilization of GPU cycles. Thus, making the system as cost effective as possible for large-scale distributed training purposes.

Data fabric delivered by NetApp overcomes the challenge by enabling data scientists and data engineers to connect together on-premises and in the cloud to have synchronous data, without performing any manual intervention. In other words, data fabric smooths the process of managing AI workflow spread across multiple locations. It also facilitates on-demand-based data availability by bringing data close to compute and performing analysis, training, and validation wherever and whenever needed. This capability not only enables data integration but also protection and security of the entire data pipeline.

Additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

- Dataset: TuSimple
  https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection

- Deep Learning Network Architecture: Spatial Convolutional Neural Network
  https://arxiv.org/abs/1712.06080

- Distributed deep learning training framework: Horovod
  https://horovod.ai/

- RUN: AI container orchestration solution: RUN: AI product introduction
  https://docs.run.ai/home/components/

- RUN: AI installation documentation
TR-4841: Hybrid Cloud AI Operating System with Data Caching

Rick Huang, David Arnette, NetApp
Yochay Ettun, cnvrg.io

The explosive growth of data and the exponential growth of ML and AI have converged to create a zettabyte economy with unique development and implementation challenges.

Although it is widely known that ML models are data-hungry and require high-performance data storage proximal to compute resources, in practice, it is not so straightforward to implement this model, especially with hybrid cloud and elastic compute instances. Massive quantities of data are usually stored in low-cost data lakes, where high-performance AI compute resources such as GPUs cannot efficiently access it. This problem is aggravated in a hybrid-cloud infrastructure where some workloads operate in the cloud and some are located on-premises or in a different HPC environment entirely.

In this document, we present a novel solution that allows IT professionals and data engineers to create a truly hybrid cloud AI platform with a topology-aware data hub that enables data scientists to instantly and automatically create a cache of their datasets in proximity to their compute resources, wherever they are.
located. As a result, not only can high-performance model training be accomplished, but additional benefits are created, including the collaboration of multiple AI practitioners, who have immediate access to dataset caches, versions, and lineages within a dataset version hub.

Use Case Overview and Problem Statement

Datasets and dataset versions are typically located in a data lake, such as NetApp StorageGrid object-based storage, which offers reduced cost and other operational advantages. Data scientists pull these datasets and engineer them in multiple steps to prepare them for training with a specific model, often creating multiple versions along the way. As the next step, the data scientist must pick optimized compute resources (GPUs, high-end CPU instances, an on-premises cluster, and so on) to run the model. The following figure depicts the lack of dataset proximity in an ML compute environment.

However, multiple training experiments must run in parallel in different compute environments, each of which require a download of the dataset from the data lake, which is an expensive and time-consuming process. Proximity of the dataset to the compute environment (especially for a hybrid cloud) is not guaranteed. In addition, other team members that run their own experiments with the same dataset must go through the same arduous process. Beyond the obvious slow data access, challenges include difficulties tracking dataset versions, dataset sharing, collaboration, and reproducibility.

Customer Requirements

Customer requirements can vary in order to achieve high-performance ML runs while efficiently using resources; for example, customers might require the following:

• Fast access to datasets from each compute instance executing the training model without incurring expensive downloads and data access complexities
• The use any compute instance (GPU or CPU) in the cloud or on-premises without concern for the location
  of the datasets
• Increased efficiency and productivity by running multiple training experiments in parallel with different
  compute resources on the same dataset without unnecessary delays and data latency
• Minimized compute instance costs
• Improved reproducibility with tools to keep records of the datasets, their lineage, versions, and other
  metadata details
• Enhanced sharing and collaboration so that any authorized member of the team can access the datasets
  and run experiments

To implement dataset caching with NetApp ONTAP data management software, customers must perform the
following tasks:

• Configure and set the NFS storage that is closest to the compute resources.
• Determine which dataset and version to cache.
• Monitor the total memory committed to cached datasets and how much NFS storage is available for
  additional cache commits (for example, cache management).
• Age out of datasets in the cache if they have not been used in certain time. The default is one day; other
  configuration options are available.

Solution Overview

This section reviews a conventional data science pipeline and its drawbacks. It also
presents the architecture of the proposed dataset caching solution.

Conventional Data Science Pipeline and Drawbacks

A typical sequence of ML model development and deployment involves iterative steps that include the
following:

• Ingesting data
• Data preprocessing (creating multiple versions of the datasets)
• Running multiple experiments involving hyperparameter optimization, different models, and so on
• Deployment
• Monitoring.cnvrg.io has developed a comprehensive platform to automate all tasks from research to
  deployment. A small sample of dashboard screenshots pertaining to the pipeline is shown in the following
  figure.
It is very common to have multiple datasets in play from public repositories and private data. In addition, each dataset is likely to have multiple versions resulting from dataset cleanup or feature engineering. A dashboard that provides a dataset hub and a version hub is needed to make sure collaboration and consistency tools are available to the team, as can be seen in the following figure.
The next step in the pipeline is training, which requires multiple parallel instances of training models, each associated with a dataset and a certain compute instance. The binding of a dataset to a certain experiment with a certain compute instance is a challenge because it is possible that some experiments are performed by GPU instances from Amazon Web Services (AWS), while other experiments are performed by DGX-1 or DGX-2 instances on-premises. Other experiments might be executed in CPU servers in GCP, while the dataset location is not in reasonable proximity to the compute resources performing the training. A reasonable proximity would have full 10GbE or more low-latency connectivity from the dataset storage to the compute instance.

It is a common practice for data scientists to download the dataset to the compute instance performing the training and execute the experiment. However, there are several potential problems with this approach:

- When the data scientist downloads the dataset to a compute instance, there are no guarantees that the integrated compute storage is high performance (an example of a high-performance system would be the ONTAP AFF A800 NVMe solution).
- When the downloaded dataset resides in one compute node, storage can become a bottleneck when distributed models are executed over multiple nodes (unlike with NetApp ONTAP high-performance distributed storage).
- The next iteration of the training experiment might be performed in a different compute instance due to queue conflicts or priorities, again creating significant network distance from the dataset to the compute location.
- Other team members executing training experiments on the same compute cluster cannot share this dataset; each performs the (expensive) download of the dataset from an arbitrary location.
- If other datasets or versions of the same dataset are needed for the subsequent training jobs, the data scientists must again perform the (expensive) download of the dataset to the compute instance performing the training. NetApp and cnvrg.io have created a new dataset caching solution that eliminates these
hurdles. The solution creates accelerated execution of the ML pipeline by caching hot datasets on the ONTAP high-performance storage system. With ONTAP NFS, the datasets are cached once (and only once) in a data fabric powered by NetApp (such as AFF A800), which is collocated with the compute. As the NetApp ONTAP NFS high-speed storage can serve multiple ML compute nodes, the performance of the training models is optimized, bringing cost savings, productivity, and operational efficiency to the organization.

Solution Architecture

This solution from NetApp and cnvrg.io provides dataset caching, as shown in the following figure. Dataset caching allows data scientists to pick a desired dataset or dataset version and move it to the ONTAP NFS cache, which lies in proximity to the ML compute cluster. The data scientist can now run multiple experiments without incurring delays or downloads. In addition, all collaborating engineers can use the same dataset with the attached compute cluster (with the freedom to pick any node) without additional downloads from the data lake. The data scientists are offered a dashboard that tracks and monitors all datasets and versions and provides a view of which datasets were cached.

The cnvrg.io platform auto-detects aged datasets that have not been used for a certain time and evicts them from the cache, which maintains free NFS cache space for more frequently used datasets. It is important to note that dataset caching with ONTAP works in the cloud and on-premises, thus providing maximum flexibility.

Concepts and Components

This section covers concepts and components associated with data caching in an ML workflow.
Machine Learning

ML is rapidly becoming essential to many businesses and organizations around the world. Therefore, IT and DevOps teams are now facing the challenge of standardizing ML workloads and provisioning cloud, on-premises, and hybrid compute resources that support the dynamic and intensive workflows that ML jobs and pipelines require.

Container-Based Machine Learning and Kubernetes

Containers are isolated user-space instances that run on top of a shared host operating system kernel. The adoption of containers is rapidly increasing. Containers offer many of the same application sandboxing benefits that virtual machines (VMs) offer. However, because the hypervisor and guest operating system layers that VMs rely on have been eliminated, containers are far more lightweight.

Containers also allow the efficient packaging of application dependencies, run times, and so on directly with an application. The most commonly used container packaging format is the Docker container. An application that has been containerized in the Docker container format can be executed on any machine that can run Docker containers. This is true even if the application’s dependencies are not present on the machine, because all dependencies are packaged in the container itself. For more information, visit the Docker website.

Kubernetes, the popular container orchestrator, allows data scientists to launch flexible, container-based jobs and pipelines. It also enables infrastructure teams to manage and monitor ML workloads in a single managed and cloud-native environment. For more information, visit the Kubernetes website.

cnvrg.io

cnvrg.io is an AI operating system that transforms the way enterprises manage, scale, and accelerate AI and data science development from research to production. The code-first platform is built by data scientists for data scientists and offers flexibility to run on-premises or in the cloud. With model management, MLOps, and continual ML solutions, cnvrg.io brings top-of-the-line technology to data science teams so they can spend less time on DevOps and focus on the real magic—algorithms. Since using cnvrg.io, teams across industries have gotten more models to production resulting in increased business value.

**cnvrg.io Meta-Scheduler**

cnvrg.io has a unique architecture that allows IT and engineers to attach different compute resources to the same control plane and have cnvrg.io manage ML jobs across all resources. This means that IT can attach multiple on-premises Kubernetes clusters, VM servers, and cloud accounts and run ML workloads on all resources, as shown in the following figure.
cnvrg.io Data Caching

cnvrg.io allows data scientists to define hot and cold dataset versions with its data-caching technology. By default, datasets are stored in a centralized object storage database. Then, data scientists can cache a specific data version on the selected compute resource to save time on download and therefore increase ML development and productivity. Datasets that are cached and are not in use for a few days are automatically cleared from the selected NFS. Caching and clearing the cache can be performed with a single click; no coding, IT, or DevOps work is required.

cnvrg.io Flows and ML Pipelines

cnvrg.io Flows is a tool for building production ML pipelines. Each component in a flow is a script/code running on a selected compute with a base docker image. This design enables data scientists and engineers to build a single pipeline that can run both on-premises and in the cloud. cnvrg.io makes sure data, parameters, and artifacts are moving between the different components. In addition, each flow is monitored and tracked for 100% reproducible data science.

cnvrg.io CORE

cnvrg.io CORE is a free platform for the data science community to help data scientists focus more on data science and less on DevOps. CORE’s flexible infrastructure gives data scientists the control to use any language, AI framework, or compute environment whether on-premises or in the cloud so they can do what they do best, build algorithms. cnvrg.io CORE can be easily installed with a single command on any Kubernetes cluster.

NetApp ONTAP AI

ONTAP AI is a data center reference architecture for ML and deep learning (DL) workloads that uses NetApp AFF storage systems and NVIDIA DGX systems with Tesla V100 GPUs. ONTAP AI is based on the industry-standard NFS file protocol over 100Gb Ethernet, providing customers with a high-performance ML/DL infrastructure that uses standard data center technologies to reduce implementation and administration overhead. Using standardized network and protocols enables ONTAP AI to integrate into hybrid cloud environments while maintaining operational consistency and simplicity. As a prevalidated infrastructure solution, ONTAP AI reduces deployment time and risk and reduces administration overhead significantly, allowing customers to realize faster time to value.

NVIDIA DeepOps

DeepOps is an open source project from NVIDIA that, by using Ansible, automates the deployment of GPU server clusters according to best practices. DeepOps is modular and can be used for various deployment tasks. For this document and the validation exercise that it describes, DeepOps is used to deploy a Kubernetes cluster that consists of GPU server worker nodes. For more information, visit the DeepOps website.

NetApp Trident

Trident is an open source storage orchestrator developed and maintained by NetApp that greatly simplifies the creation, management, and consumption of persistent storage for Kubernetes workloads. Trident itself a Kubernetes-native application—it runs directly within a Kubernetes cluster. With Trident, Kubernetes users (developers, data scientists, Kubernetes administrators, and so on) can create, manage, and interact with persistent storage volumes in the standard Kubernetes format that they are already familiar with. At the same time, they can take advantage of NetApp advanced data management capabilities and a data fabric that is powered by NetApp technology. Trident abstracts away the complexities of persistent storage and makes it simple to consume. For more information, visit the Trident website.
NetApp StorageGRID

NetApp StorageGRID is a software-defined object storage platform designed to meet these needs by providing simple, cloud-like storage that users can access using the S3 protocol. StorageGRID is a scale-out system designed to support multiple nodes across internet-connected sites, regardless of distance. With the intelligent policy engine of StorageGRID, users can choose erasure-coding objects across sites for geo-resiliency or object replication between remote sites to minimize WAN access latency. StorageGrid provides an excellent private-cloud primary object storage data lake in this solution.

NetApp Cloud Volumes ONTAP

NetApp Cloud Volumes ONTAP data management software delivers control, protection, and efficiency to user data with the flexibility of public cloud providers including AWS, Google Cloud Platform, and Microsoft Azure. Cloud Volumes ONTAP is cloud-native data management software built on the NetApp ONTAP storage software, providing users with a superior universal storage platform that addresses their cloud data needs. Having the same storage software in the cloud and on-premises provides users with the value of a data fabric without having to train IT staff in all-new methods to manage data.

For customers that are interested in hybrid cloud deployment models, Cloud Volumes ONTAP can provide the same capabilities and class-leading performance in most public clouds to provide a consistent and seamless user experience in any environment.

Hardware and Software Requirements

This section covers the technology requirements for the ONTAP AI solution.

Hardware Requirements

Although hardware requirements depend on specific customer workloads, ONTAP AI can be deployed at any scale for data engineering, model training, and production inferencing from a single GPU up to rack-scale configurations for large-scale ML/DL operations. For more information about ONTAP AI, see the ONTAP AI website.

This solution was validated using a DGX-1 system for compute, a NetApp AFF A800 storage system, and Cisco Nexus 3232C for network connectivity. The AFF A800 used in this validation can support as many as 10 DGX-1 systems for most ML/DL workloads. The following figure shows the ONTAP AI topology used for model training in this validation.
To extend this solution to a public cloud, Cloud Volumes ONTAP can be deployed alongside cloud GPU compute resources and integrated into a hybrid cloud data fabric that enables customers to use whatever resources are appropriate for any given workload.

Software Requirements

The following table shows the specific software versions used in this solution validation.

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ubuntu</td>
<td>18.04.4 LTS</td>
</tr>
<tr>
<td>NVIDIA DGX OS</td>
<td>4.4.0</td>
</tr>
<tr>
<td>NVIDIA DeepOps</td>
<td>20.02.1</td>
</tr>
<tr>
<td>Kubernetes</td>
<td>1.15</td>
</tr>
<tr>
<td>Helm</td>
<td>3.1.0</td>
</tr>
<tr>
<td>cnvrg.io</td>
<td>3.0.0</td>
</tr>
<tr>
<td>NetApp ONTAP</td>
<td>9.6P4</td>
</tr>
</tbody>
</table>

For this solution validation, Kubernetes was deployed as a single-node cluster on the DGX-1 system. For large-scale deployments, independent Kubernetes master nodes should be deployed to provide high availability of management services as well as reserve valuable DGX resources for ML and DL workloads.

Solution Deployment and Validation Details

The following sections discuss the details of solution deployment and validation.

ONTAP AI Deployment

Deployment of ONTAP AI requires the installation and configuration of networking, compute, and storage hardware. Specific instructions for deployment of the ONTAP AI infrastructure are beyond the scope of this document. For detailed deployment
information, see NVA-1121-DEPLOY: NetApp ONTAP AI, Powered by NVIDIA.

For this solution validation, a single volume was created and mounted to the DGX-1 system. That mount point was then mounted to the containers to make data accessible for training. For large-scale deployments, NetApp Trident automates the creation and mounting of volumes to eliminate administrative overhead and enable end-user management of resources.

**Kubernetes Deployment**

To deploy and configure your Kubernetes cluster with NVIDIA DeepOps, perform the following tasks from a deployment jump host:

1. Download NVIDIA DeepOps by following the instructions on the [Getting Started page](#) on the NVIDIA DeepOps GitHub site.
2. Deploy Kubernetes in your cluster by following the instructions on the [Kubernetes Deployment Guide](#) on the NVIDIA DeepOps GitHub site.

   For the DeepOps Kubernetes deployment to work, the same user must exist on all Kubernetes master and worker nodes.

   If the deployment fails, change the value of `kubectl_localhost` to false in `deepops/config/group_vars/k8s-cluster.yml` and repeat step 2. The Copy `kubectl` binary to ansible host task, which executes only when the value of `kubectl_localhost` is true, relies on the fetch Ansible module, which has known memory usage issues. These memory usage issues can sometimes cause the task to fail. If the task fails because of a memory issue, then the remainder of the deployment operation does not complete successfully.

   If the deployment completes successfully after you have changed the value of `kubectl_localhost` to false, then you must manually copy the `kubectl` binary from a Kubernetes master node to the deployment jump host. You can find the location of the `kubectl` binary on a specific master node by running the `which kubectl` command directly on that node.

**cnvrg.io Deployment**

**Deploy cnvrg CORE Using Helm**

Helm is the easiest way to quickly deploy cnvrg using any cluster, on-premises, Minikube, or on any cloud cluster (such as AKS, EKS, and GKE). This section describes how cnvrg was installed on an on-premises (DGX-1) instance with Kubernetes installed.

**Prerequisites**

Before you can complete the installation, you must install and prepare the following dependencies on your local machine:

- Kubectl
- Helm 3.x
- Kubernetes cluster 1.15+
Deploy Using Helm

1. To download the most updated cnvrg helm charts, run the following command:

   ```
   helm repo add cnvrg https://helm.cnvrg.io
   helm repo update
   ```

2. Before you deploy cnvrg, you need the external IP address of the cluster and the name of the node on which you will deploy cnvrg. To deploy cnvrg on an on-premises Kubernetes cluster, run the following command:

   ```
   helm install cnvrg cnvrg/cnvrg --timeout 1500s --wait \ --set
global.external_ip=<ip_of_cluster> \ --set global.node=<name_of_node>
   ```

3. Run the `helm install` command. All the services and systems automatically install on your cluster. The process can take up to 15 minutes.

4. The `helm install` command can take up to 10 minutes. When the deployment completes, go to the URL of your newly deployed cnvrg or add the new cluster as a resource inside your organization. The `helm` command informs you of the correct URL.

   ```
   Thank you for installing cnvrg.io!
   Your installation of cnvrg.io is now available, and can be reached via:
   Talk to our team via email at
   ```

5. When the status of all the containers is running or complete, cnvrg has been successfully deployed. It should look similar to the following example output:

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>cnvrg-app-69fbb9df98-6xrgf</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
<tr>
<td>cnvrg-sidekiq-b9d54d889-5x4fc</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
<tr>
<td>controller-65895b47d4-s96v6</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
<tr>
<td>init-app-vs-config-wv9c4</td>
<td>0/1</td>
<td>Completed</td>
<td>0</td>
<td>9m</td>
</tr>
<tr>
<td>init-gateway-vs-config-2zbpp</td>
<td>0/1</td>
<td>Completed</td>
<td>0</td>
<td>9m</td>
</tr>
<tr>
<td>init-minio-vs-config-cd2rg</td>
<td>0/1</td>
<td>Completed</td>
<td>0</td>
<td>9m</td>
</tr>
<tr>
<td>minio-0</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
<tr>
<td>postgres-0</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
<tr>
<td>redis-695c49c986-kcbt9</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
<tr>
<td>seeder-wh655</td>
<td>0/1</td>
<td>Completed</td>
<td>0</td>
<td>2m</td>
</tr>
<tr>
<td>speaker-5sghr</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
</tbody>
</table>

Computer Vision Model Training with ResNet50 and the Chest X-ray Dataset

cnvrg.io AI OS was deployed on a Kubernetes setup on a NetApp ONTAP AI architecture powered by the NVIDIA DGX system. For validation, we used the NIH Chest X-ray dataset consisting of de-identified images of
chest x-rays. The images were in the PNG format. The data was provided by the NIH Clinical Center and is available through the NIH download site. We used a 250GB sample of the data with 627,615 images across 15 classes.

The dataset was uploaded to the cnvrg platform and was cached on an NFS export from the NetApp AFF A800 storage system.

Set up the Compute Resources

The cnvrg architecture and meta-scheduling capability allow engineers and IT professionals to attach different compute resources to a single platform. In our setup, we used the same cluster cnvrg that was deployed for running the deep-learning workloads. If you need to attach additional clusters, use the GUI, as shown in the following screenshot.

![cnvrg platform screenshot](image)

Load Data

To upload data to the cnvrg platform, you can use the GUI or the cnvrg CLI. For large datasets, NetApp recommends using the CLI because it is a strong, scalable, and reliable tool that can handle a large number of files.

To upload data, complete the following steps:

1. Download the **cnvrg CLI**.
2. Navigate to the x-ray directory.
3. Initialize the dataset in the platform with the `cnvrg data init` command.
4. Upload all contents of the directory to the central data lake with the `cnvrg data sync` command. After the data is uploaded to the central object store (StorageGRID, S3, or others), you can browse with the GUI. The following figure shows a loaded chest X-ray fibrosis image PNG file. In addition, cnvrg versions the data so that any model you build can be reproduced down to the data version.
Cach Data

To make training faster and avoid downloading 600k+ files for each model training and experiment, we used the data-caching feature after data was initially uploaded to the central data-lake object store.
After users click Cache, cnvrg downloads the data in its specific commit from the remote object store and caches it on the ONTAP NFS volume. After it completes, the data is available for instant training. In addition, if the data is not used for a few days (for model training or exploration, for example), cnvrg automatically clears the cache.

**Build an ML Pipeline with Cached Data**

cnvrg flows allows you to easily build production ML pipelines. Flows are flexible, can work for any kind of ML use case, and can be created through the GUI or code. Each component in a flow can run on a different compute resource with a different Docker image, which makes it possible to build hybrid cloud and optimized ML pipelines.
Building the Chest X-ray Flow: Setting Data

We added our dataset to a newly created flow. When adding the dataset, you can select the specific version (commit) and indicate whether you want the cached version. In this example, we selected the cached commit.
Building the Chest X-ray Flow: Setting Training Model: ResNet50

In the pipeline, you can add any kind of custom code you want. In cnvrg, there is also the AI library, a reusable ML components collection. In the AI library, there are algorithms, scripts, data sources, and other solutions that can be used in any ML or deep learning flow. In this example, we selected the prebuilt ResNet50 module. We used default parameters such as batch_size:128, epochs:10, and more. These parameters can be viewed in the AI Library docs. The following screenshot shows the new flow with the X-ray dataset connected to ResNet50.
Define the Compute Resource for ResNet50

Each algorithm or component in cnvrg flows can run on a different compute instance, with a different Docker image. In our setup, we wanted to run the training algorithm on the NVIDIA DGX systems with the NetApp ONTAP AI architecture. In the following figure, we selected `gpu-real`, which is a compute template and specification for our on-premises cluster. We also created a queue of templates and selected multiple templates. In this way, if the `gpu-real` resource cannot be allocated (if, for example, other data scientists are using it), then you can enable automatic cloud-bursting by adding a cloud provider template. The following screenshot shows the use of `gpu-real` as a compute node for ResNet50.
Tracking and Monitoring Results

After a flow is executed, cnvrg triggers the tracking and monitoring engine. Each run of a flow is automatically documented and updated in real time. Hyperparameters, metrics, resource usage (GPU utilization, and more), code version, artifacts, logs, and so on are automatically available in the Experiments section, as shown in the following two screenshots.
Conclusion

NetApp and cnvrg.io have partnered to offer customers a complete data management solution for ML and DL software development. ONTAP AI provides high-performance compute and storage for any scale of operation, and cnvrg.io software streamlines data science workflows and improves resource utilization.

Acknowledgments

- Mike Oglesby, Technical Marketing Engineer, NetApp
- Santosh Rao, Senior Technical Director, NetApp

Where to Find Additional Information

To learn more about the information that is described in this document, see the following resources:

- Cnvrg.io (https://cnvrg.io):
  - Cnvrg CORE (free ML platform)
    https://cnvrg.io/platform/core
  - Cnvrg docs
    https://app.cnvrg.io/docs

- NVIDIA DGX-1 servers:
  - NVIDIA DGX-1 servers
  - NVIDIA Tesla V100 Tensor Core GPU
  - NVIDIA GPU Cloud (NGC)

- NetApp AFF systems:
  - AFF datasheet
  - NetApp FlashAdvantage for AFF
  - ONTAP 9.x documentation
• NetApp FlexGroup technical report

• NetApp persistent storage for containers:
 ◦ NetApp Trident
    https://netapp.io/persistent-storage-provisioner-for-kubernetes/

• NetApp Interoperability Matrix:
  ◦ NetApp Interoperability Matrix Tool
    http://support.netapp.com/matrix

• ONTAP AI networking:
  ◦ Cisco Nexus 3232C Switches
  ◦ Mellanox Spectrum 2000 series switches

• ML framework and tools:
  ◦ DALI
    https://github.com/NVIDIA/DALI
    https://www.tensorflow.org/
  ◦ Horovod: Uber’s Open-Source Distributed Deep Learning Framework for TensorFlow
    https://eng.uber.com/horovod/
  ◦ Enabling GPUs in the Container Runtime Ecosystem
    https://devblogs.nvidia.com/gpu-containers-runtime/
  ◦ Docker
    https://docs.docker.com
  ◦ Kubernetes
    https://kubernetes.io/docs/home/
  ◦ NVIDIA DeepOps
    https://github.com/NVIDIA/deepops
  ◦ Kubeflow
Several emerging application scenarios, such as advanced driver-assistance systems (ADAS), Industry 4.0, smart cities, and Internet of Things (IoT), require the processing of continuous data streams under a near-zero latency. This document describes a compute and storage architecture to deploy GPU-based artificial intelligence (AI) inferencing on NetApp storage controllers and Lenovo ThinkSystem servers in an edge environment that meets these requirements. This document also provides performance data for the industry standard MLPerf Inference benchmark, evaluating various inference tasks on edge servers equipped with NVIDIA T4 GPUs. We investigate the performance of offline, single stream, and multistream inference scenarios and show that the architecture with a cost-effective shared networked storage system is highly performant and provides a central point for data and model management for multiple edge servers.

Introduction

Companies are increasingly generating massive volumes of data at the network edge. To achieve maximum value from smart sensors and IoT data, organizations are looking for a real-time event streaming solution that enables edge computing. Computationally demanding jobs are therefore increasingly performed at the edge, outside of data centers. AI inference is one of the drivers of this trend. Edge servers provide sufficient computational power for these workloads, especially when using accelerators, but limited storage is often an issue, especially in multiserver environments. In this document we show how you can deploy a shared storage system in the edge environment and how it benefits AI inference workloads without imposing a performance penalty.

This document describes a reference architecture for AI inference at the edge. It combines multiple Lenovo ThinkSystem edge servers with a NetApp storage system to create a solution that is easy to deploy and manage. It is intended to be a baseline guide for practical deployments in various situations, such as the factory floor with multiple cameras and industrial sensors, point-of-sale (POS) systems in retail transactions, or Full Self-Driving (FSD) systems that identify visual anomalies in autonomous vehicles.

This document covers testing and validation of a compute and storage configuration consisting of Lenovo ThinkSystem SE350 Edge Server and an entry-level NetApp AFF and EF-Series storage system. The reference architectures provide an efficient and cost-effective solution for AI deployments while also providing...
comprehensive data services, integrated data protection, seamless scalability, and cloud connected data storage with NetApp ONTAP and NetApp SANtricity data management software.

**Target audience**

This document is intended for the following audiences:

- Business leaders and enterprise architects who want to productize AI at the edge.
- Data scientists, data engineers, AI/machine learning (ML) researchers, and developers of AI systems.
- Enterprise architects who design solutions for the development of AI/ML models and applications.
- Data scientists and AI engineers looking for efficient ways to deploy deep learning (DL) and ML models.
- Edge device managers and edge server administrators responsible for deployment and management of edge inferencing models.

**Solution architecture**

This Lenovo ThinkSystem server and NetApp ONTAP or NetApp SANtricity storage solution is designed to handle AI inferencing on large datasets using the processing power of GPUs alongside traditional CPUs. This validation demonstrates high performance and optimal data management with an architecture that uses either single or multiple Lenovo SR350 edge servers interconnected with a single NetApp AFF storage system, as shown in the following two figures.
The logical architecture overview in the following figure shows the roles of the compute and storage elements in this architecture. Specifically, it shows the following:

- Edge compute devices performing inference on the data it receives from cameras, sensors, and so on.
- A shared storage element that serves multiple purposes:
  - Provides a central location for inference models and other data needed to perform the inference. Compute servers access the storage directly and use inference models across the network without the need to copy them locally.
  - Updated models are pushed here.
  - Archives input data that edge servers receive for later analysis. For example, if the edge devices are connected to cameras, the storage element keeps the videos captured by the cameras.
This NetApp and Lenovo solution offers the following key benefits:

- GPU accelerated computing at the edge.
- Deployment of multiple edge servers backed and managed from a shared storage.
- Robust data protection to meet low recovery point objectives (RPOs) and recovery time objectives (RTOs) with no data loss.
- Optimized data management with NetApp Snapshot copies and clones to streamline development workflows.

**How to use this architecture**

This document validates the design and performance of the proposed architecture. However, we have not tested certain software-level pieces, such as container, workload, or model management and data synchronization with cloud or data center on-premises, because they are specific to a deployment scenario. Here, multiple choices exist.

At the container management level, Kubernetes container management is a good choice and is well supported in either a fully upstream version (Canonical) or in a modified version suitable for enterprise deployments (Red Hat). The NetApp AI Control Plane which uses NetApp Trident and the newly added NetApp DataOps Toolkit provides built-in traceability, data management functions, interfaces, and tools for data scientists and data engineers to integrate with NetApp storage. Kubeflow, the ML toolkit for Kubernetes, provides additional AI capabilities along with a support for model versioning and KFServing on several platforms such as TensorFlow Serving or NVIDIA Triton Inference Server. Another option is NVIDIA EGX platform, which provides workload management along with access to a catalog of GPU-enabled AI inference containers. However, these options might require significant effort and expertise to put them into production and might require the assistance of a third-party independent software vendor (ISV) or consultant.

**Solution areas**

The key benefit of AI inferencing and edge computing is the ability of devices to compute, process, and analyze data with a high level of quality without latency. There are far too many examples of edge computing use cases to describe in this document, but here are a few prominent ones:
Automobiles: Autonomous vehicles

The classic edge computing illustration is in the advanced driver-assistance systems (ADAS) in autonomous vehicles (AV). The AI in driverless cars must rapidly process a lot of data from cameras and sensors to be a successful safe driver. Taking too long to interpret between an object and a human can mean life or death, therefore being able to process that data as close to the vehicle as possible is crucial. In this case, one or more edge compute servers handles the input from cameras, RADAR, LiDAR, and other sensors, while shared storage holds inference models and stores input data from sensors.

Healthcare: Patient monitoring

One of the greatest impacts of AI and edge computing is its ability to enhance continuous monitoring of patients for chronic diseases both in at-home care and intensive care units (ICUs). Data from edge devices that monitor insulin levels, respiration, neurological activity, cardiac rhythm, and gastrointestinal functions require instantaneous analysis of data that must be acted on immediately because there is limited time to act to save someone’s life.

Retail: Cashier-less payment

Edge computing can power AI and ML to help retailers reduce checkout time and increase foot traffic. Cashier-less systems support various components, such as the following:

- Authentication and access. Connecting the physical shopper to a validated account and permitting access to the retail space.
- Inventory monitoring. Using sensors, RFID tags, and computer vision systems to help confirm the selection or deselection of items by shoppers.

Here, each of the edge servers handle each checkout counter and the shared storage system serves as a central synchronization point.

Financial services: Human safety at kiosks and fraud prevention

Banking organizations are using AI and edge computing to innovate and create personalized banking experiences. Interactive kiosks using real-time data analytics and AI inferencing now enable ATMs to not only help customers withdraw money, but proactively monitor kiosks through the images captured from cameras to identify risk to human safety or fraudulent behavior. In this scenario, edge compute servers and shared storage systems are connected to interactive kiosks and cameras to help banks collect and process data with AI inference models.

Manufacturing: Industry 4.0

The fourth industrial revolution (Industry 4.0) has begun, along with emerging trends such as Smart Factory and 3D printing. To prepare for a data-led future, large-scale machine-to-machine (M2M) communication and IoT are integrated for increased automation without the need for human intervention. Manufacturing is already highly automated and adding AI features is a natural continuation of the long-term trend. AI enables automating operations that can be automated with the help of computer vision and other AI capabilities. You can automate quality control or tasks that rely on human vision or decision making to perform faster analyses of materials on assembly lines in factory floors to help manufacturing plants meet the required ISO standards of safety and quality management. Here, each compute edge server is connected to an array of sensors monitoring the manufacturing process and updated inference models are pushed to the shared storage, as needed.
Telecommunications: Rust detection, tower inspection, and network optimization

The telecommunications industry uses computer vision and AI techniques to process images that automatically detect rust and identify cell towers that contain corrosion and, therefore, require further inspection. The use of drone images and AI models to identify distinct regions of a tower to analyze rust, surface cracks, and corrosion has increased in recent years. The demand continues to grow for AI technologies that enable telecommunication infrastructure and cell towers to be inspected efficiently, assessed regularly for degradation, and repaired promptly when required.

Additionally, another emerging use case in telecommunication is the use of AI and ML algorithms to predict data traffic patterns, detect 5G-capable devices, and automate and augment multiple-input and multiple-output (MIMO) energy management. MIMO hardware is used at radio towers to increase network capacity; however, this comes with additional energy costs. ML models for “MIMO sleep mode” deployed at cell sites can predict the efficient use of radios and help reduce energy consumption costs for mobile network operators (MNOs). AI inferencing and edge computing solutions help MNOs reduce the amount of data transmitted back-and-forth to data centers, lower their TCO, optimize network operations, and improve overall performance for end users.

Technology overview

This section describes the technological foundation for this AI solution.

NetApp AFF systems

State-of-the-art NetApp AFF storage systems enable AI inference deployments at the edge to meet enterprise storage requirements with industry-leading performance, superior flexibility, cloud integration, and best-in-class data management. Designed specifically for flash, NetApp AFF systems help accelerate, manage, and protect business-critical data.

- Entry-level NetApp AFF storage systems are based on FAS2750 hardware and SSD flash media
- Two controllers in HA configuration

NetApp entry-level AFF C190 storage systems support the following features:

- A maximum drive count of 24x 960GB SSDs
- Two possible configurations:
- Ethernet (10GbE): 4x 10GBASE-T (RJ-45) ports
- Unified (16Gb FC or 10GbE): 4x unified target adapter 2 (UTA2) ports
- A maximum of 50.5TB effective capacity

For NAS workloads, a single entry-level AFF C190 system supports throughput of 4.4GBps for sequential reads and 230K IOPS for small random reads at latencies of 1ms or less.

**NetApp AFF A220**

NetApp also offers other entry-level storage systems that provide higher performance and scalability for larger-scale deployments. For NAS workloads, a single entry-level AFF A220 system supports:

- Throughput of 6.2GBps for sequential reads
- 375K IOPS for small random reads at latencies of 1ms or less
- Maximum drive count of 144x 960GB, 3.8TB, or 7.6TB SSDs
- AFF A220 scales to larger than 1PB of effective capacity

**NetApp AFF A250**

- Maximum effective capacity is 35PB with maximum scale out 2-24 nodes (12 HA pairs)
- Provides ≥ 45% performance increase over AFF A220
- 440k IOPS random reads @1ms
- Built on the latest NetApp ONTAP release: ONTAP 9.8
- Leverages two 25Gb Ethernet for HA and cluster interconnect

**NetApp E-Series EF Systems**

The EF-Series is a family of entry-level and mid-range all-flash SAN storage arrays that can accelerate access to your data and help you derive value from it faster with NetApp SANtricity software. These systems offer both SAS and NVMe flash storage and provide you with affordable to extreme IOPS, response times under 100 microseconds, and bandwidth up to 44GBps—making them ideal for mixed workloads and demanding applications such as AI inferencing and high-performance computing (HPC).

The following figure shows the NetApp EF280 storage system.
NetApp EF280

- 32Gb/16Gb FC, 25Gb/10Gb iSCSI, and 12Gb SAS support
- Maximum effective capacity is 96 drives totaling 1.5PB
- Throughput of 10GBps (sequential reads)
- 300K IOPs (random reads)
- The NetApp EF280 is the lowest cost all-flash array (AFA) in the NetApp portfolio

NetApp EF300

- 24x NVMe SSD drives for a total capacity of 367TB
- Expansion options totaling 240x NL-SAS HDDs, 96x SAS SSDs, or a combination
- 100Gb NVMe/IB, NVMe/RoCE, iSER/IB, and SRP/IB
- 32Gb NVME/FC, FCP
- 25Gb iSCSI
- 20GBps (sequential reads)
- 670K IOPs (random reads)

For more information, see the NetApp EF-Series NetApp EF-Series all-flash arrays EF600, F300, EF570, and EF280 datasheet.

NetApp ONTAP 9

ONTAP 9.8.1, the latest generation of storage management software from NetApp, enables businesses to modernize infrastructure and transition to a cloud-ready data center. Leveraging industry-leading data management capabilities, ONTAP enables the management and protection of data with a single set of tools, regardless of where that data resides. You can also move data freely to wherever it is needed: the edge, the core, or the cloud. ONTAP 9.8.1 includes numerous features that simplify data management, accelerate and protect critical data, and enable next generation infrastructure capabilities across hybrid cloud architectures.

Simplify data management

Data management is crucial to enterprise IT operations so that appropriate resources are used for applications and datasets. ONTAP includes the following features to streamline and simplify operations and reduce the total cost of operation:

- **Inline data compaction and expanded deduplication.** Data compaction reduces wasted space inside storage blocks, and deduplication significantly increases effective capacity. This applies to data stored locally and data tiered to the cloud.

- **Minimum, maximum, and adaptive quality of service (AQoS).** Granular quality of service (QoS) controls help maintain performance levels for critical applications in highly shared environments.

- **NetApp FabricPool.** This feature provides automatic tiering of cold data to public and private cloud storage options, including Amazon Web Services (AWS), Azure, and NetApp StorageGRID storage solution. For more information about FabricPool, see TR-4598.

Accelerate and protect data

ONTAP 9 delivers superior levels of performance and data protection and extends these capabilities in the following ways:
• **Performance and lower latency.** ONTAP offers the highest possible throughput at the lowest possible latency.

• **Data protection.** ONTAP provides built-in data protection capabilities with common management across all platforms.

• **NetApp Volume Encryption (NVE).** ONTAP offers native volume-level encryption with both onboard and External Key Management support.

• **Multitenancy and multifactor authentication.** ONTAP enables sharing of infrastructure resources with the highest levels of security.

**Future-proof infrastructure**

ONTAP 9 helps meet demanding and constantly changing business needs with the following features:

• **Seamless scaling and nondisruptive operations.** ONTAP supports the nondisruptive addition of capacity to existing controllers and to scale-out clusters. Customers can upgrade to the latest technologies, such as NVMe and 32Gb FC, without costly data migrations or outages.

• **Cloud connection.** ONTAP is the most cloud-connected storage management software, with options for software-defined storage (ONTAP Select) and cloud-native instances (NetApp Cloud Volumes Service) in all public clouds.

• **Integration with emerging applications.** ONTAP offers enterprise-grade data services for next generation platforms and applications, such as autonomous vehicles, smart cities, and Industry 4.0, by using the same infrastructure that supports existing enterprise apps.

**NetApp SANtricity**

NetApp SANtricity is designed to deliver industry-leading performance, reliability, and simplicity to E-Series hybrid-flash and EF-Series all-flash arrays. Achieve maximum performance and utilization of your E-Series hybrid-flash and EF-Series all-flash arrays for heavy-workload applications, including data analytics, video surveillance, and backup and recovery. With SANtricity, configuration tweaking, maintenance, capacity expansion, and other tasks can be completed while the storage stays online. SANtricity also provides superior data protection, proactive monitoring, and certified security—all accessible through the easy-to-use, on-box System Manager interface. To learn more, see the NetApp E-Series SANtricity Software datasheet.

**Performance optimized**

Performance-optimized SANtricity software delivers data—with high IOPs, high throughput, and low latency—to all your data analytics, video surveillance, and backup apps. Accelerate performance for high-IOPS, low-latency applications and high-bandwidth, high-throughput applications.

**Maximize uptime**

Complete all your management tasks while the storage stays online. Tweak configurations, perform maintenance, or expand capacity without disrupting I/O. Realize best-in-class reliability with automated features, online configuration, state-of-the-art Dynamic Disk Pools (DPP) technology, and more.

**Rest easy**

SANtricity software delivers superior data protection, proactive monitoring, and certified security—all through the easy-to-use, on-box System Manager interface. Simplify storage-management chores. Gain the flexibility you need for advanced tuning of all E-Series storage systems. Manage your NetApp E-Series system—anytime, anywhere. Our on-box, web-based interface streamlines your management workflow.
NetApp Trident

Trident from NetApp is an open-source dynamic storage orchestrator for Docker and Kubernetes that simplifies the creation, management, and consumption of persistent storage. Trident, a Kubernetes native application, runs directly within a Kubernetes cluster. Trident enables customers to seamlessly deploy DL container images onto NetApp storage and provides an enterprise-grade experience for AI container deployments. Kubernetes users (such as ML developers and data scientists) can create, manage, and automate orchestration and cloning to take advantage of NetApp advanced data management capabilities powered by NetApp technology.

NetApp BlueXP Copy and Sync

BlueXP Copy and Sync is a NetApp service for rapid and secure data synchronization. Whether you need to transfer files between on-premises NFS or SMB file shares, NetApp StorageGRID, NetApp ONTAP S3, NetApp Cloud Volumes Service, Azure NetApp Files, Amazon Simple Storage Service (Amazon S3), Amazon Elastic File System (Amazon EFS), Azure Blob, Google Cloud Storage, or IBM Cloud Object Storage, BlueXP Copy and Sync moves the files where you need them quickly and securely. After your data is transferred, it is fully available for use on both source and target. BlueXP Copy and Sync continuously synchronizes the data, based on your predefined schedule, moving only the deltas, so time and money spent on data replication is minimized. BlueXP Copy and Sync is a software as a service (SaaS) tool that is extremely simple to set up and use. Data transfers that are triggered by BlueXP Copy and Sync are carried out by data brokers. You can deploy BlueXP Copy and Sync data brokers in AWS, Azure, Google Cloud Platform, or on-premises.

Lenovo ThinkSystem servers

Lenovo ThinkSystem servers feature innovative hardware, software, and services that solve customers’ challenges today and deliver an evolutionary, fit-for-purpose, modular design approach to address tomorrow’s challenges. These servers capitalize on best-in-class, industry-standard technologies coupled with differentiated Lenovo innovations to provide the greatest possible flexibility in x86 servers.

Key advantages of deploying Lenovo ThinkSystem servers include:

- Highly scalable, modular designs to grow with your business
- Industry-leading resilience to save hours of costly unscheduled downtime
- Fast flash technologies for lower latencies, quicker response times, and smarter data management in real time

In the AI area, Lenovo is taking a practical approach to helping enterprises understand and adopt the benefits of ML and AI for their workloads. Lenovo customers can explore and evaluate Lenovo AI offerings in Lenovo AI Innovation Centers to fully understand the value for their particular use case. To improve time to value, this customer-centric approach gives customers proof of concept for solution development platforms that are ready to use and optimized for AI.

Lenovo ThinkSystem SE350 Edge Server

Edge computing allows data from IoT devices to be analyzed at the edge of the network before being sent to the data center or cloud. The Lenovo ThinkSystem SE350, as shown in the figure below, is designed for the unique requirements for deployment at the edge, with a focus on flexibility, connectivity, security, and remote manageability in a compact ruggedized and environmentally hardened form factor.

Featuring the Intel Xeon D processor with the flexibility to support acceleration for edge AI workloads, the SE350 is purpose-built for addressing the challenge of server deployments in a variety of environments outside the data center.
MLPerf

MLPerf is the industry-leading benchmark suite for evaluating AI performance. It covers many areas of applied AI including image classification, object detection, medical imaging, and natural language processing (NLP). In this validation, we used Inference v0.7 workloads, which is the latest iteration of the MLPerf Inference at the completion of this validation. The MLPerf Inference v0.7 suite includes four new benchmarks for data center and edge systems:

- **BERT.** Bi-directional Encoder Representation from Transformers (BERT) fine-tuned for question answering by using the SQuAD dataset.
- **DLRM.** Deep Learning Recommendation Model (DLRM) is a personalization and recommendation model that is trained to optimize click-through rates (CTR).
- **3D U-Net.** 3D U-Net architecture is trained on the Brain Tumor Segmentation (BraTS) dataset.
- **RNN-T.** Recurrent Neural Network Transducer (RNN-T) is an automatic speech recognition (ASR) model
that is trained on a subset of LibriSpeech. MLPerf Inference results and code are publicly available and released under Apache license. MLPerf Inference has an Edge division, which supports the following scenarios:

- **Single stream.** This scenario mimics systems where responsiveness is a critical factor, such as offline AI queries performed on smartphones. Individual queries are sent to the system and response times are recorded. 90th percentile latency of all the responses is reported as the result.

- **Multistream.** This benchmark is for systems that process input from multiple sensors. During the test, queries are sent at a fixed time interval. A QoS constraint (maximum allowed latency) is imposed. The test reports the number of streams that the system can process while meeting the QoS constraint.

- **Offline.** This is the simplest scenario covering batch processing applications and the metric is throughput in samples per second. All data is available to the system and the benchmark measures the time it takes to process all the samples.

Lenovo has published MLPerf Inference scores for SE350 with T4, the server used in this document. See the results at [https://mlperf.org/inference-results-0-7/](https://mlperf.org/inference-results-0-7/) in the “Edge, Closed Division” section in entry #0.7-145.

**Test plan**

This document follows MLPerf Inference v0.7 code, MLPerf Inference v1.1 code, and rules. We ran MLPerf benchmarks designed for inference at the edge as defined in the following table.

<table>
<thead>
<tr>
<th>Area</th>
<th>Task</th>
<th>Model</th>
<th>Dataset</th>
<th>QSL size</th>
<th>Quality</th>
<th>Multistream latency constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vision</td>
<td>Image classification</td>
<td>Resnet50v1.5</td>
<td>ImageNet (224x224)</td>
<td>1024</td>
<td>99% of FP32</td>
<td>50ms</td>
</tr>
<tr>
<td>Vision</td>
<td>Object detection (large)</td>
<td>SSD-ResNet34</td>
<td>COCO (1200x1200)</td>
<td>64</td>
<td>99% of FP32</td>
<td>66ms</td>
</tr>
<tr>
<td>Vision</td>
<td>Object detection (small)</td>
<td>SSD-MobileNetsv1</td>
<td>COCO (300x300)</td>
<td>256</td>
<td>99% of FP32</td>
<td>50ms</td>
</tr>
<tr>
<td>Vision</td>
<td>Medical image segmentation</td>
<td>3D UNET</td>
<td>BraTS 2019 (224x224x160)</td>
<td>16</td>
<td>99% and 99.9% of FP32</td>
<td>n/a</td>
</tr>
<tr>
<td>Speech</td>
<td>Speech-to-text</td>
<td>RNNT</td>
<td>Librispeech dev-clean</td>
<td>2513</td>
<td>99% of FP32</td>
<td>n/a</td>
</tr>
<tr>
<td>Language</td>
<td>Language processing</td>
<td>BERT</td>
<td>SQuAD v1.1</td>
<td>10833</td>
<td>99% of FP32</td>
<td>n/a</td>
</tr>
</tbody>
</table>

The following table presents Edge benchmark scenarios.

<table>
<thead>
<tr>
<th>Area</th>
<th>Task</th>
<th>Scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vision</td>
<td>Image classification</td>
<td>Single stream, offline, multistream</td>
</tr>
<tr>
<td>Vision</td>
<td>Object detection (large)</td>
<td>Single stream, offline, multistream</td>
</tr>
</tbody>
</table>

286
We performed these benchmarks using the networked storage architecture developed in this validation and compared results to those from local runs on the edge servers previously submitted to MLPerf. The comparison is to determine how much impact the shared storage has on inference performance.

Test configuration

The following figure shows the test configuration. We used the NetApp AFF C190 storage system and two Lenovo ThinkSystem SE350 servers (each with one NVIDIA T4 accelerator). These components are connected through a 10GbE network switch. The network storage holds validation/test datasets and pretrained models. The servers provide computational capability, and the storage is accessed over NFS protocol.

This section describes the tested configurations, the network infrastructure, the SE350 server, and the storage provisioning details. The following table lists the base components for the solution architecture.

<table>
<thead>
<tr>
<th>Solution components</th>
<th>Details</th>
</tr>
</thead>
</table>
| Lenovo ThinkSystem servers | • 2x SE350 servers each with one NVIDIA T4 GPU card
• Each server contains one Intel Xeon D-2123IT CPU with four physical cores running at 2.20GHz and 128GB RAM |
| Entry-level NetApp AFF storage system (HA pair) | • NetApp ONTAP 9 software
• 24x 960GB SSDs
• NFS protocol
• One interface group per controller, with four logical IP addresses for mount points |
The following table lists the storage configuration: AFF C190 with 2RU, 24 drive slots.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Aggregate</th>
<th>FlexGroup volume</th>
<th>Aggregatesize</th>
<th>Volumesize</th>
<th>Operating systemmount point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller1</td>
<td>Aggr1</td>
<td>/netapplenovo_AI_fg</td>
<td>8.42TiB</td>
<td>15TB</td>
<td>/netapp_lenovo_fg</td>
</tr>
<tr>
<td>Controller2</td>
<td>Aggr2</td>
<td></td>
<td>8.42TiB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The /netappLenovo_AI_fg folder contains the datasets used for model validation.

The figure below shows the test configuration. We used the NetApp EF280 storage system and two Lenovo ThinkSystem SE350 servers (each with one NVIDIA T4 accelerator). These components are connected through a 10GbE network switch. The network storage holds validation/test datasets and pretrained models. The servers provide computational capability, and the storage is accessed over NFS protocol.

The following table lists the storage configuration for EF280.
## Test procedure

This section describes the test procedures used to validate this solution.

### Operating system and AI inference setup

For AFF C190, we used Ubuntu 18.04 with NVIDIA drivers and docker with support for NVIDIA GPUs and used MLPerf code available as a part of the Lenovo submission to MLPerf Inference v0.7.

For EF280, we used Ubuntu 20.04 with NVIDIA drivers and docker with support for NVIDIA GPUs and MLPerf code available as a part of the Lenovo submission to MLPerf Inference v1.1.

To set up the AI inference, follow these steps:

1. Download datasets that require registration, the ImageNet 2012 Validation set, Criteo Terabyte dataset, and BraTS 2019 Training set, and then unzip the files.
2. Create a working directory with at least 1TB and define environmental variable `MLPERF_SCRATCH_PATH` referring to the directory.

You should share this directory on the shared storage for the network storage use case, or the local disk when testing with local data.
3. Run the make `prebuild` command, which builds and launches the docker container for the required inference tasks.

   The following commands are all executed from within the running docker container:

   - Download pretrained AI models for MLPerf Inference tasks: `make download_model`
   - Download additional datasets that are freely downloadable: `make download_data`
   - Preprocess the data: `make preprocess_data`
   - Run: `make build`.
   - Build inference engines optimized for the GPU in compute servers: `make generate_engines`
   - To run Inference workloads, run the following (one command):

   ```
   make run_harness RUN_ARGS="--benchmarks=<BENCHMARKS> --scenarios=<SCENARIOS>
   ```

**AI inference runs**

Three types of runs were executed:

- Single server AI inference using local storage
- Single server AI inference using network storage
- Multi-server AI inference using network storage

**Test results**

A multitude of tests were run to evaluate the performance of the proposed architecture.

There are six different workloads (image classification, object detection [small], object detection [large], medical imaging, speech-to-text, and natural language processing [NLP]), which you can run in three different scenarios: offline, single stream, and multistream.

The last scenario is implemented only for image classification and object detection.

This gives 15 possible workloads, which were all tested under three different setups:

- Single server/local storage
- Single server/network storage
- Multi-server/network storage

The results are described in the following sections.

**AI inference in offline scenario for AFF**

In this scenario, all the data was available to the server and the time it took to process all the samples was measured. We report bandwidths in samples per second as the results of the tests. When more than one compute server was used, we report total bandwidth summed over all the servers. The results for all three use
cases are shown in the figure below. For the two-server case, we report combined bandwidth from both servers.

The results show that network storage does not negatively affect the performance—the change is minimal and for some tasks, none is found. When adding the second server, the total bandwidth either exactly doubles, or at worst, the change is less than 1%.

**AI inference in a single stream scenario for AFF**

This benchmark measures latency. For the multiple computational server case, we report the average latency. The results for the suite of tasks are given in the figure below. For the two-server case, we report the average latency from both servers.
The results, again, show that the network storage is sufficient to handle the tasks. The difference between local and network storage in the one server case is minimal or none. Similarly, when two servers use the same storage, the latency on both servers stays the same or changes by a very small amount.

**AI inference in multistream scenario for AFF**

In this case, the result is the number of streams that the system can handle while satisfying the QoS constraint. Thus, the result is always an integer. For more than one server, we report the total number of streams summed over all the servers. Not all workloads support this scenario, but we have executed those that do. The results of our tests are summarized in the figure below. For the two-server case, we report the combined number of streams from both servers.
The results show perfect performance of the setup—local and networking storage give the same results and adding the second server doubles the number of streams the proposed setup can handle.

Test results for EF

A multitude of tests were run to evaluate the performance of the proposed architecture. There are six different workloads (image classification, object detection [small], object detection [large], medical imaging, speech-to-text, and natural language processing [NLP]), which were run in two different scenarios: offline and single stream. The results are described in the following sections.

**AI inference in offline scenario for EF**

In this scenario, all the data was available to the server and the time it took to process all the samples was measured. We report bandwidths in samples per second as the results of the tests. For single node runs we report average from both servers, while for two server runs we report total bandwidth summed over all the servers. The results for use cases are shown in the figure below.
The results show that network storage does not negatively affect the performance—the change is minimal and for some tasks, none is found. When adding the second server, the total bandwidth either exactly doubles, or at worst, the change is less than 1%.

**AI inference in a single stream scenario for EF**

This benchmark measures latency. For all cases, we report average latency across all servers involved in the runs. The results for the suite of tasks are given.
The results show again that the network storage is sufficient to handle the tasks. The difference between the local and network storage in the one server case is minimal or none. Similarly, when two servers use the same storage, the latency on both servers stays the same or changes by a very small amount.

**Architecture sizing options**

You can adjust the setup used for the validation to fit other use cases.

**Compute server**

We used an Intel Xeon D-2123IT CPU, which is the lowest level of CPU supported in SE350, with four physical cores and 60W TDP. While the server does not support replacing CPUs, it can be ordered with a more powerful CPU. The top CPU supported is Intel Xeon D-2183IT with 16 cores, 100W running at 2.20GHz. This increases the CPU computational capability considerably. While CPU was not a bottleneck for running the inference workloads themselves, it helps with data processing and other tasks related to inference. At present, NVIDIA T4 is the only GPU available for edge use cases; therefore, currently, there is no ability to upgrade or downgrade the GPU.

**Shared storage**

For testing and validation, the NetApp AFF C190 system, which has maximum storage capacity of 50.5TB, a throughput of 4.4GBps for sequential reads, and 230K IOPS for small random reads, was used for the purpose of this document and is proven to be well-suited for edge inference workloads.

However, if you require more storage capacity or faster networking speeds, you should use the NetApp AFF A220 or NetApp AFF A250 storage systems. In addition, the NetApp EF280 system, which has a maximum capacity of 1.5PB, bandwidth 10GBps was also used for the purpose of this solution validation. If you prefer more storage capacity with higher bandwidth, NetApp EF300 can be used.
Conclusion

AI-driven automation and edge computing is a leading approach to help business organizations achieve digital transformation and maximize operational efficiency and safety. With edge computing, data is processed much faster because it does not have to travel to and from a data center. Therefore, the cost associated with sending data back and forth to data centers or the cloud is diminished. Lower latency and increased speed can be beneficial when businesses must make decisions in near-real time using AI inferencing models deployed at the edge.

NetApp storage systems deliver the same or better performance as local SSD storage and offer the following benefits to data scientists, data engineers, AI/ML developers, and business or IT decision makers:

- Effortless sharing of data between AI systems, analytics, and other critical business systems. This data sharing reduces infrastructure overhead, improves performance, and streamlines data management across the enterprise.
- Independently scalable compute and storage to minimize costs and improve resource usage.
- Streamlined development and deployment workflows using integrated Snapshot copies and clones for instantaneous and space-efficient user workspaces, integrated version control, and automated deployment.
- Enterprise-grade data protection for disaster recovery and business continuity. The NetApp and Lenovo solution presented in this document is a flexible, scale-out architecture that is ideal for enterprise-grade AI inference deployments at the edge.

Acknowledgments

- J.J. Falkanger, Sr. Manager, HPC & AI Solutions, Lenovo
- Dave Arnette, Technical Marketing Engineer, NetApp
- Joey Parnell, Tech Lead E-Series AI Solutions, NetApp
- Cody Harryman, QA Engineer, NetApp

Where to find additional information

To learn more about the information described in this document, refer to the following documents and/or websites:

- NetApp AFF A-Series arrays product page
  https://www.netapp.com/data-storage/aff-a-series/
- NetApp ONTAP data management software—ONTAP 9 information library
- TR-4727: NetApp EF-Series Introduction
- NetApp E-Series SANtricity Software Datasheet
The NVIDIA DGX family of systems is made up of the world’s first integrated artificial intelligence (AI)-based systems that are purpose-built for enterprise AI. NetApp AFF storage systems deliver extreme performance and industry-leading hybrid cloud data-management capabilities. NetApp and NVIDIA have partnered to create the NetApp ONTAP AI reference architecture, a turnkey solution for AI and machine learning (ML) workloads that provides enterprise-class performance, reliability, and support.

This white paper gives directional guidance to customers building conversational AI systems in support of different use cases in various industry verticals. It includes information about the deployment of the system using NVIDIA Jarvis. The tests were performed using an NVIDIA DGX Station and a NetApp AFF A220 storage system.

The target audience for the solution includes the following groups:

- Enterprise architects who design solutions for the development of AI models and software for conversational AI use cases such as a virtual retail assistant
- Data scientists looking for efficient ways to achieve language modeling development goals
- Data engineers in charge of maintaining and processing text data such as customer questions and dialogue transcripts
- Executive and IT decision makers and business leaders interested in transforming the conversational AI experience and achieving the fastest time to market from AI initiatives
Solution Overview

NetApp ONTAP AI and BlueXP Copy and Sync

The NetApp ONTAP AI architecture, powered by NVIDIA DGX systems and NetApp cloud-connected storage systems, was developed and verified by NetApp and NVIDIA. This reference architecture gives IT organizations the following advantages:

- Eliminates design complexities
- Enables independent scaling of compute and storage
- Enables customers to start small and scale seamlessly
- Offers a range of storage options for various performance and cost points

NetApp ONTAP AI tightly integrates DGX systems and NetApp AFF A220 storage systems with state-of-the-art networking. NetApp ONTAP AI and DGX systems simplify AI deployments by eliminating design complexity and guesswork. Customers can start small and grow their systems in an uninterrupted manner while intelligently managing data from the edge to the core to the cloud and back.

NetApp BlueXP Copy and Sync enables you to move data easily over various protocols, whether it’s between two NFS shares, two CIFS shares, or one file share and Amazon S3, Amazon Elastic File System (EFS), or Azure Blob storage. Active-active operation means that you can continue to work with both source and target at the same time, incrementally synchronizing data changes when required. By enabling you to move and incrementally synchronize data between any source and destination system, whether on-premises or cloud-based, BlueXP Copy and Sync opens up a wide variety of new ways in which you can use data. Migrating data between on-premises systems, cloud on-boarding and cloud migration, or collaboration and data analytics all become easily achievable. The figure below shows available sources and destinations.

In conversational AI systems, developers can leverage BlueXP Copy and Sync to archive conversation history from the cloud to data centers to enable offline training of natural language processing (NLP) models. By training models to recognize more intents, the conversational AI system will be better equipped to manage more complex questions from end-users.

NVIDIA Jarvis Multimodal Framework
NVIDIA Jarvis is an end-to-end framework for building conversational AI services. It includes the following GPU-optimized services:

- Automatic speech recognition (ASR)
- Natural language understanding (NLU)
- Integration with domain-specific fulfillment services
- Text-to-speech (TTS)
- Computer vision (CV)

Jarvis-based services use state-of-the-art deep learning models to address the complex and challenging task of real-time conversational AI. To enable real-time, natural interaction with an end user, the models need to complete computation in under 300 milliseconds. Natural interactions are challenging, requiring multimodal sensory integration. Model pipelines are also complex and require coordination across the above services.

Jarvis is a fully accelerated, application framework for building multimodal conversational AI services that use an end-to-end deep learning pipeline. The Jarvis framework includes pretrained conversational AI models, tools, and optimized end-to-end services for speech, vision, and NLU tasks. In addition to AI services, Jarvis enables you to fuse vision, audio, and other sensor inputs simultaneously to deliver capabilities such as multi-user, multi-context conversations in applications such as virtual assistants, multi-user diarization, and call center assistants.

NVIDIA NeMo

NVIDIA NeMo is an open-source Python toolkit for building, training, and fine-tuning GPU-accelerated state-of-the-art conversational AI models using easy-to-use application programming interfaces (APIs). NeMo runs mixed precision compute using Tensor Cores in NVIDIA GPUs and can scale up to multiple GPUs easily to deliver the highest training performance possible. NeMo is used to build models for real-time ASR, NLP, and TTS applications such as video call transcriptions, intelligent video assistants, and automated call center support across different industry verticals, including healthcare, finance, retail, and telecommunications.

We used NeMo to train models that recognize complex intents from user questions in archived conversation history. This training extends the capabilities of the retail virtual assistant beyond what Jarvis supports as delivered.

Retail Use Case Summary

Using NVIDIA Jarvis, we built a virtual retail assistant that accepts speech or text input and answers questions regarding weather, points-of-interest, and inventory pricing. The conversational AI system is able to remember conversation flow, for example, ask a follow-up question if the user does not specify location for weather or points-of-interest. The system also recognizes complex entities such as “Thai food” or “laptop memory.” It understands natural language questions like “will it rain next week in Los Angeles?” A demonstration of the retail virtual assistant can be found in Customize States and Flows for Retail Use Case.

Solution Technology

The following figure illustrates the proposed conversational AI system architecture. You can interact with the system with either speech signal or text input. If spoken input is detected, Jarvis AI-as-service (AIaaS) performs ASR to produce text for Dialog Manager. Dialog Manager remembers states of conversation, routes text to corresponding services, and passes commands to Fulfillment Engine. Jarvis NLP Service takes in text, recognizes intents and entities, and outputs those intents and entity slots back to Dialog Manager, which then sends Action to Fulfillment Engine. Fulfillment Engine consists of third-party
APIs or SQL databases that answer user queries. After receiving Result from Fulfillment Engine, Dialog Manager routes text to Jarvis TTS AlaaS to produce an audio response for the end-user. We can archive conversation history, annotate sentences with intents and slots for NeMo training such that NLP Service improves as more users interact with the system.

Hardware Requirements

This solution was validated using one DGX Station and one AFF A220 storage system. Jarvis requires either a T4 or V100 GPU to perform deep neural network computations.

The following table lists the hardware components that are required to implement the solution as tested.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>T4 or V100 GPU</td>
<td>1</td>
</tr>
<tr>
<td>NVIDIA DGX Station</td>
<td>1</td>
</tr>
</tbody>
</table>

Software Requirements

The following table lists the software components that are required to implement the solution as tested.

<table>
<thead>
<tr>
<th>Software</th>
<th>Version or Other Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetApp ONTAP data management software</td>
<td>9.6</td>
</tr>
<tr>
<td>Cisco NX-OS switch firmware</td>
<td>7.0(3)I6(1)</td>
</tr>
<tr>
<td>NVIDIA DGX OS</td>
<td>4.0.4 - Ubuntu 18.04 LTS</td>
</tr>
<tr>
<td>NVIDIA Jarvis Framework</td>
<td>EA v0.2</td>
</tr>
</tbody>
</table>
Overview

This section provides detail on the implementation of the virtual retail assistant.

Jarvis Deployment

You can sign up for Jarvis Early Access program to gain access to Jarvis containers on NVIDIA GPU Cloud (NGC). After receiving credentials from NVIDIA, you can deploy Jarvis using the following steps:

1. Sign-on to NGC.
2. Set your organization on NGC: ea-2-jarvis.
3. Locate Jarvis EA v0.2 assets: Jarvis containers are in Private Registry > Organization Containers.
4. Select Jarvis: navigate to Model Scripts and click Jarvis Quick Start
5. Verify that all assets are working properly.
6. Find the documentation to build your own applications: PDFs can be found in Model Scripts > Jarvis Documentation > File Browser.

Customize States and Flows for Retail Use Case

You can customize States and Flows of Dialog Manager for your specific use cases. In our retail example, we have the following four yaml files to direct the conversation according to different intents.

See the following list of file names and description of each file:

- main_flow.yml: Defines the main conversation flows and states and directs the flow to the other three yaml files when necessary.
- retail_flow.yml: Contains states related to retail or points-of-interest questions. The system either provides the information of the nearest store, or the price of a given item.
- weather_flow.yml: Contains states related to weather questions. If the location cannot be determined, the system asks a follow up question to clarify.
- error_flow.yml: Handles cases where user intents do not fall into the above three yaml files. After displaying an error message, the system re-routes back to accepting user questions. The following sections contain the detailed definitions for these yaml files.

main_flow.yml

```yaml
name: JarvisRetail
intent_transitions:
```
jarvis_error: error
price_check: retail_price_check
inventory_check: retail_inventory_check
store_location: retail_store_location
weather.weather: weather
weather.temperature: temperature
weather.sunny: sunny
weather.cloudy: cloudy
weather.snow: snow
weather.rainfall: rain
weather.snow_yes_no: snowfall
weather.rainfall_yes_no: rainfall
weather.temperature_yes_no: tempyesno
weather.humidity: humidity
weather.humidity_yes_no: humidity
navigation.startnavigationpoi: retail # Transitions should be context
and slot based. Redirecting for now.
navigation.geteta: retail
navigation.showdirection: retail
navigation.showmappoi: idk_what_you_talkin_about
nomatch.none: idk_what_you_talkin_about
states:
  init:
    type: message_text
    properties:
      text: "Hi, welcome to NARA retail and weather service. How can I help you?"
  input_intent:
    type: input_context
    properties:
      nlp_type: jarvis
      entities:
        intent: dontcare
    # This state is executed if the intent was not understood
  dont_get_the_intent:
    type: message_text_random
    properties:
      responses:
        - "Sorry I didn't get that! Please come again."
        - "I beg your pardon! Say that again?"
        - "Are we talking about weather? What would you like to know?"
        - "Sorry I know only about the weather"
        - "You can ask me about the weather, the rainfall, the temperature, I don't know much more"
    delay: 0
    transitions:
idk_what_you_talkin_about:
  type: message_text_random
  properties:
  responses:
  - "Sorry I didn't get that! Please come again."
  - "I beg your pardon! Say that again?"
  - "Are we talking about retail or weather? What would you like to know?"
  - "Sorry I know only about retail and the weather"
  - "You can ask me about retail information or the weather, the rainfall, the temperature. I don't know much more."
  delay: 0
  transitions:
    next_state: input_intent
error:
  type: change_context
  properties:
    update_keys:
      intent: 'error'
  transitions:
    flow: error_flow
retail_inventory_check:
  type: change_context
  properties:
    update_keys:
      intent: 'retail_inventory_check'
  transitions:
    flow: retail_flow
retail_price_check:
  type: change_context
  properties:
    update_keys:
      intent: 'check_item_price'
  transitions:
    flow: retail_flow
retail_store_location:
  type: change_context
  properties:
    update_keys:
      intent: 'find_the_store'
  transitions:
    flow: retail_flow
weather:
  type: change_context
  properties:
update_keys:
  intent: 'weather'
transitions:
  flow: weather_flow

temperature:
  type: change_context
  properties:
    update_keys:
      intent: 'temperature'
  transitions:
    flow: weather_flow

rainfall:
  type: change_context
  properties:
    update_keys:
      intent: 'rainfall'
  transitions:
    flow: weather_flow

sunny:
  type: change_context
  properties:
    update_keys:
      intent: 'sunny'
  transitions:
    flow: weather_flow

cloudy:
  type: change_context
  properties:
    update_keys:
      intent: 'cloudy'
  transitions:
    flow: weather_flow

snow:
  type: change_context
  properties:
    update_keys:
      intent: 'snow'
  transitions:
    flow: weather_flow

rain:
  type: change_context
  properties:
    update_keys:
      intent: 'rain'
  transitions:
    flow: weather_flow
snowfall:
  type: change_context
  properties:
    update_keys:
      intent: 'snowfall'
  transitions:
    flow: weather_flow

tempyesno:
  type: change_context
  properties:
    update_keys:
      intent: 'tempyesno'
  transitions:
    flow: weather_flow

humidity:
  type: change_context
  properties:
    update_keys:
      intent: 'humidity'
  transitions:
    flow: weather_flow

day_state:
  type: reset
  transitions:
    next_state: init

retail_flow.yml

name: retail_flow
states:
  store_location:
    type: conditional_exists
    properties:
      key: '{{location}}'
    transitions:
      exists: retail_state
      notexists: ask_retail_location
  retail_state:
    type: Retail
    properties:
    transitions:
      next_state: output_retail
  output_retail:
    type: message_text
    properties:
ask_retail_location:
  type: message_text
  properties:
    text: "For which location? I can find the closest store near you."
  transitions:
    next_state: input_retail_location

input_retail_location:
  type: input_user
  properties:
    nlp_type: jarvis
    entities:
      slot: location
      require_match: true
  transitions:
    match: retail_state
    notmatch: check_retail_jarvis_error

output_retail_acknowledge:
  type: message_text_random
  properties:
    responses:
      - 'ok in {{location}}'
      - 'the store in {{location}}'
      - 'I always wanted to shop in {{location}}'
    delay: 0
  transitions:
    next_state: retail_state

output_retail_notlocation:
  type: message_text
  properties:
    text: "I did not understand the location. Can you please repeat?"
  transitions:
    next_state: input_intent

check_retail_jarvis_error:
  type: conditional_exists
  properties:
    key: '{{jarvis_error}}'
  transitions:
    exists: show_retail_jarvis_api_error
    notexists: output_retail_notlocation

show_retail_jarvis_api_error:
  type: message_text
  properties:
    text: "I am having trouble understanding right now. Come again on
that?

transitions:
  next_state: input_intent

weather_flow.yml

name: weather_flow
states:
  check_weather_location:
    type: conditional_exists
    properties:
      key: '{{location}}'
    transitions:
      exists: weather_state
      notexists: ask_weather_location
  weather_state:
    type: Weather
    properties:
    transitions:
      next_state: output_weather
  output_weather:
    type: message_text
    properties:
      text: '{{weather_status}}'
    transitions:
      next_state: input_intent
  ask_weather_location:
    type: message_text
    properties:
      text: "For which location?"
    transitions:
      next_state: input_weather_location
  input_weather_location:
    type: input_user
    properties:
      nlp_type: jarvis
      entities:
        slot: location
        require_match: true
    transitions:
      match: weather_state
      notmatch: check_jarvis_error
  output_weather_acknowledge:
    type: message_text_random
    properties:
responses:
  - 'ok in {{location}}'
  - 'the weather in {{location}}'
  - 'I always wanted to go in {{location}}'
delay: 0
transitions:
  next_state: weather_state
output_weather_notlocation:
  type: message_text
  properties:
    text: "I did not understand the location, can you please repeat?"
  transitions:
    next_state: input_intent
check_jarvis_error:
  type: conditional_exists
  properties:
    key: '{{jarvis_error}}'
  transitions:
    exists: show_jarvis_api_error
    notexists: output_weather_notlocation
show_jarvis_api_error:
  type: message_text
  properties:
    text: "I am having troubled understanding right now. Come again on that, else check jarvis services?"
  transitions:
    next_state: input_intent
name: error_flow
states:
  error_state:
    type: message_text_random
    properties:
      responses:
        - "Sorry I didn't get that!"
        - "Are we talking about retail or weather? What would you like to know?"
        - "Sorry I know only about retail information or the weather"
        - "You can ask me about retail information or the weather, the rainfall, the temperature. I don't know much more"
        - "Let's talk about retail or the weather!"
    delay: 0
    transitions:
      next_state: input_intent

Connect to Third-Party APIs as Fulfillment Engine

We connected the following third-party APIs as a Fulfillment Engine to answer questions:

- WeatherStack API: returns weather, temperature, rainfall, and snow in a given location.
- Yelp Fusion API: returns the nearest store information in a given location.
- eBay Python SDK: returns the price of a given item.

NetApp Retail Assistant Demonstration

We recorded a demonstration video of NetApp Retail Assistant (NARA).

Video demonstration of NARA
Use NetApp BlueXP Copy and Sync to Archive Conversation History

By dumping conversation history into a CSV file once a day, we can then leverage BlueXP Copy and Sync to download the log files into local storage. The following figure shows the architecture of having Jarvis deployed on-premises and in public clouds, while using BlueXP Copy and Sync to send conversation history for NeMo training. Details of NeMo training can be found in the section Expand Intent Models Using NeMo Training.
Expand Intent Models Using NeMo Training

NVIDIA NeMo is a toolkit built by NVIDIA for creating conversational AI applications. This toolkit includes collections of pre-trained modules for ASR, NLP, and TTS, enabling researchers and data scientists to easily compose complex neural network architectures and put more focus on designing their own applications.

As shown in the previous example, NARA can only handle a limited type of question. This is because the pre-trained NLP model only trains on these types of questions. If we want to enable NARA to handle a broader range of questions, we need to retrain it with our own datasets. Thus, here, we demonstrate how we can use NeMo to extend the NLP model to satisfy the requirements. We start by converting the log collected from NARA into the format for NeMo, and then train with the dataset to enhance the NLP model.

Model

Our goal is to enable NARA to sort the items based on user preferences. For instance, we might ask NARA to suggest the highest-rated sushi restaurant or might want NARA to look up the jeans with the lowest price. To this end, we use the intent detection and slot filling model provided in NeMo as our training model. This model allows NARA to understand the intent of searching preference.

Data Preparation

To train the model, we collect the dataset for this type of question, and convert it to the NeMo format. Here, we listed the files we use to train the model.

**dict.intents.csv**

This file lists all the intents we want the NeMo to understand. Here, we have two primary intents and one intent only used to categorize the questions that do not fit into any of the primary intents.

```
price_check
find_the_store
unknown
```

**dict.slots.csv**

This file lists all the slots we can label on our training questions.

```
B-store.type
B-store.name
B-store.status
B-store.hour.start
B-store.hour.end
B-store.hour.day
B-item.type
B-item.name
B-item.color
B-item.size
B-item.quantity
```
This is the main training dataset. Each line starts with the question following the intent category listing in the file
dict.intent.csv. The label is enumerated starting from zero.

**train_slots.tsv**

```
20 46 24 25 6 32 6
52 52 24 6
23 52 14 40 52 25 6 32 6
...
```

**Train the Model**

```
docker pull nvcr.io/nvidia/nemo:v0.10
```

We then use the following command to launch the container. In this command, we limit the container to use a single GPU (GPU ID = 1) since this is a lightweight training exercise. We also map our local workspace `/workspace/nemo/` to the folder inside container `/nemo`.

```
NV_GPU='1' docker run --runtime=nvidia -it --shm-size=16g \
  --network=host --ulimit memlock=-1 --ulimit stack=67108864 \
  -v /workspace/nemo:/nemo\n  --rm nvcr.io/nvidia/nemo:v0.10
```

Inside the container, if we want to start from the original pre-trained BERT model, we can use the following command to start the training procedure. `data_dir` is the argument to set up the path of the training data. `work_dir` allows you to configure where you want to store the checkpoint files.

```
    cd examples/nlp/intent_detection_slot_tagging/\n    python joint_intent_slot_with_bert.py \n        --data_dir /nemo/training_data\n        --work_dir /nemo/log
```

If we have new training datasets and want to improve the previous model, we can use the following command to continue from the point we stopped. `checkpoint_dir` takes the path to the previous checkpoints folder.

```
    cd examples/nlp/intent_detection_slot_tagging/\n    python joint_intent_slot_infer.py \n        --data_dir /nemo/training_data \n        --checkpoint_dir /nemo/log/2020-05-04_18-34-20/checkpoints/ \n        --eval_file_prefix test
```
Inference the Model

We need to validate the performance of the trained model after a certain number of epochs. The following command allows us to test the query one-by-one. For instance, in this command, we want to check if our model can properly identify the intention of the query *where can I get the best pasta*.

```bash
cd examples/nlp/intent_detection_slot_tagging/
python joint_intent_slot_infer_bl.py \
--checkpoint_dir /nemo/log/2020-05-29_23-50-58/checkpoints/ \
--query "where can i get the best pasta" \
--data_dir /nemo/training_data/ \
--num_epochs=50
```

Then, the following is the output from the inference. In the output, we can see that our trained model can properly predict the intention *find_the_store*, and return the keywords we are interested in. With these keywords, we enable the NARA to search for what users want and do a more precise search.

```
[NeMo I 2020-05-30 00:06:54 actions:728] Evaluating batch 0 out of 1
[NeMo I 2020-05-30 00:06:55 inference_utils:34] Query: where can i get the best pasta
[NeMo I 2020-05-30 00:06:55 inference_utils:36] Predicted intent:       1
find_the_store
[NeMo I 2020-05-30 00:06:55 inference_utils:50] where B-interrogative.location
[NeMo I 2020-05-30 00:06:55 inference_utils:50] can O
[NeMo I 2020-05-30 00:06:55 inference_utils:50] i O
[NeMo I 2020-05-30 00:06:55 inference_utils:50] get B-verb
[NeMo I 2020-05-30 00:06:55 inference_utils:50] the B-article
[NeMo I 2020-05-30 00:06:55 inference_utils:50] best B-rating.high
[NeMo I 2020-05-30 00:06:55 inference_utils:50] pasta B-item.type
```

Conclusion

A true conversational AI system engages in human-like dialogue, understands context, and provides intelligent responses. Such AI models are often huge and highly complex. With NVIDIA GPUs and NetApp storage, massive, state-of-the-art language models can be trained and optimized to run inference rapidly. This is a major stride towards ending the trade-off between an AI model that is fast versus one that is large and complex. GPU-optimized language understanding models can be integrated into AI applications for industries such as healthcare, retail, and financial services, powering advanced digital voice assistants in smart speakers and customer service lines. These high-quality conversational AI systems allow businesses across verticals to provide previously unattainable personalized services when engaging with customers.

Jarvis enables the deployment of use cases such as virtual assistants, digital avatars, multimodal sensor fusion (CV fused with ASR/NLP/TTS), or any ASR/NLP/TTS/CV stand-alone use case, such as transcription.
We built a virtual retail assistant that can answer questions regarding weather, points-of-interest, and inventory pricing. We also demonstrated how to improve the natural language understanding capabilities of the conversational AI system by archiving conversation history using BlueXP Copy and Sync and training NeMo models on new data.

Acknowledgments

The authors gratefully acknowledge the contributions that were made to this white paper by our esteemed colleagues from NVIDIA: Davide Onofrio, Alex Qi, Sicong Ji, Marty Jain, and Robert Sohigian. The authors would also like to acknowledge the contributions of key NetApp team members: Santosh Rao, David Arnette, Michael Oglesby, Brent Davis, Andy Sayare, Erik Mulder, and Mike McNamara.

Our sincere appreciation and thanks go to all these individuals, who provided insight and expertise that greatly assisted in the creation of this paper.

Where to Find Additional Information

To learn more about the information that is described in this document, see the following resources:

- NVIDIA DGX Station, V100 GPU, GPU Cloud
  - NVIDIA DGX Station  
  - NVIDIA V100 Tensor Core GPU  
  - NVIDIA NGC  
- NVIDIA Jarvis Multimodal Framework
  - NVIDIA Jarvis  
    https://developer.nvidia.com/nvidia-jarvis
  - NVIDIA Jarvis Early Access  
    https://developer.nvidia.com/nvidia-jarvis-early-access
- NVIDIA NeMo
  - NVIDIA NeMo  
    https://developer.nvidia.com/nvidia-nemo
  - Developer Guide  
    https://nvidia.github.io/NeMo/
- NetApp AFF systems
  - NetApp AFF A-Series Datasheet  
  - NetApp Flash Advantage for All Flash FAS  
  - ONTAP 9 Information Library  
TR-4858: NetApp Orchestration Solution with Run:AI

Rick Huang, David Arnette, Sung-Han Lin, NetApp
Yaron Goldberg, Run:AI

NetApp AFF storage systems deliver extreme performance and industry-leading hybrid cloud data-management capabilities. NetApp and Run:AI have partnered to demonstrate the unique capabilities of the NetApp ONTAP AI solution for artificial intelligence (AI) and machine learning (ML) workloads that provides enterprise-class performance, reliability, and support. Run:AI orchestration of AI workloads adds a Kubernetes-based scheduling and resource utilization platform to help researchers manage and optimize GPU utilization. Together with the NVIDIA DGX systems, the combined solution from NetApp, NVIDIA, and Run:AI provide an infrastructure stack that is purpose-built for enterprise AI workloads. This technical report gives directional guidance to customers building conversational AI systems in support of various use cases and industry verticals. It includes information about the deployment of Run:AI and a NetApp AFF A800 storage system and serves as a reference architecture for the simplest way to achieve fast, successful deployment of AI initiatives.

The target audience for the solution includes the following groups:

- Enterprise architects who design solutions for the development of AI models and software for Kubernetes-based use cases such as containerized microservices
- Data scientists looking for efficient ways to achieve efficient model development goals in a cluster environment with multiple teams and projects
- Data engineers in charge of maintaining and running production models
- Executive and IT decision makers and business leaders who would like to create the optimal Kubernetes cluster resource utilization experience and achieve the fastest time to market from AI initiatives

**Solution Overview**

**NetApp ONTAP AI and Al Control Plane**

The NetApp ONTAP AI architecture, developed and verified by NetApp and NVIDIA, is powered by NVIDIA DGX systems and NetApp cloud-connected storage systems. This reference architecture gives IT
organizations the following advantages:

- Eliminates design complexities
- Enables independent scaling of compute and storage
- Enables customers to start small and scale seamlessly
- Offers a range of storage options for various performance and cost points

NetApp ONTAP AI tightly integrates DGX systems and NetApp AFF A800 storage systems with state-of-the-art networking. NetApp ONTAP AI and DGX systems simplify AI deployments by eliminating design complexity and guesswork. Customers can start small and grow their systems in an uninterrupted manner while intelligently managing data from the edge to the core to the cloud and back.

NetApp AI Control Plane is a full stack AI, ML, and deep learning (DL) data and experiment management solution for data scientists and data engineers. As organizations increase their use of AI, they face many challenges, including workload scalability and data availability. NetApp AI Control Plane addresses these challenges through functionalities, such as rapidly cloning a data namespace just as you would a Git repo, and defining and implementing AI training workflows that incorporate the near-instant creation of data and model baselines for traceability and versioning. With NetApp AI Control Plane, you can seamlessly replicate data across sites and regions and swiftly provision Jupyter Notebook workspaces with access to massive datasets.

Run:AI Platform for AI Workload Orchestration

Run:AI has built the world’s first orchestration and virtualization platform for AI infrastructure. By abstracting workloads from the underlying hardware, Run:AI creates a shared pool of GPU resources that can be dynamically provisioned, enabling efficient orchestration of AI workloads and optimized use of GPUs. Data scientists can seamlessly consume massive amounts of GPU power to improve and accelerate their research while IT teams retain centralized, cross-site control and real-time visibility over resource provisioning, queuing, and utilization. The Run:AI platform is built on top of Kubernetes, enabling simple integration with existing IT and data science workflows.

The Run:AI platform provides the following benefits:

- **Faster time to innovation.** By using Run:AI resource pooling, queueing, and prioritization mechanisms together with a NetApp storage system, researchers are removed from infrastructure management hassles and can focus exclusively on data science. Run:AI and NetApp customers increase productivity by running as many workloads as they need without compute or data pipeline bottlenecks.

- **Increased team productivity.** Run:AI fairness algorithms guarantee that all users and teams get their fair share of resources. Policies around priority projects can be preset, and the platform enables dynamic allocation of resources from one user or team to another, helping users to get timely access to coveted GPU resources.

- **Improved GPU utilization.** The Run:AI Scheduler enables users to easily make use of fractional GPUs, integer GPUs, and multiple nodes of GPUs for distributed training on Kubernetes. In this way, AI workloads run based on your needs, not capacity. Data science teams are able to run more AI experiments on the same infrastructure.

Solution Technology

This solution was implemented with one NetApp AFF A800 system, two DGX-1 servers, and two Cisco Nexus 3232C 100GbE-switches. Each DGX-1 server is connected to the Nexus switches with four 100GbE connections that are used for inter-GPU communications by using remote direct memory access (RDMA) over Converged
Ethernet (RoCE). Traditional IP communications for NFS storage access also occur on these links. Each storage controller is connected to the network switches by using four 100GbE-links. The following figure shows the ONTAP AI solution architecture used in this technical report for all testing scenarios.

Hardware Used in This Solution

This solution was validated using the ONTAP AI reference architecture two DGX-1 nodes and one AFF A800 storage system. See NVA-1121 for more details about the infrastructure used in this validation.

The following table lists the hardware components that are required to implement the solution as tested.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGX-1 systems</td>
<td>2</td>
</tr>
<tr>
<td>AFF A800</td>
<td>1</td>
</tr>
<tr>
<td>Nexus 3232C switches</td>
<td>2</td>
</tr>
</tbody>
</table>

Software Requirements

This solution was validated using a basic Kubernetes deployment with the Run:AI operator installed. Kubernetes was deployed using the NVIDIA DeepOps deployment engine, which deploys all required components for a production-ready environment. DeepOps automatically deployed NetApp Trident for persistent storage integration with the k8s environment, and default storage classes were created so containers leverage storage from the AFF A800 storage system. For more information on Trident with Kubernetes on ONTAP AI, see TR-4798.

The following table lists the software components that are required to implement the solution as tested.
### Additional software requirements for Run:AI can be found at [Run:AI GPU cluster prerequisites](#).  

### Optimal Cluster and GPU Utilization with Run:AI  

The following sections provide details on the Run:AI installation, test scenarios, and results performed in this validation.

We validated the operation and performance of this system by using industry standard benchmark tools, including TensorFlow benchmarks. The ImageNet dataset was used to train ResNet-50, which is a famous Convolutional Neural Network (CNN) DL model for image classification. ResNet-50 delivers an accurate training result with a faster processing time, which enabled us to drive a sufficient demand on the storage.

#### Run:AI Installation  

To install Run:AI, complete the following steps:

1. Install the Kubernetes cluster using DeepOps and configure the NetApp default storage class.
2. Prepare GPU nodes:
   a. Verify that NVIDIA drivers are installed on GPU nodes.
   b. Verify that `nvidia-docker` is installed and configured as the default docker runtime.
3. Install Run:AI:
   a. Log into the **Run:AI Admin UI** to create the cluster.
   b. Download the created `runai-operator-<clustername>.yaml` file.
   c. Apply the operator configuration to the Kubernetes cluster.

   ```bash
   kubectl apply -f runai-operator-<clustername>.yaml
   ```

4. Verify the installation:
   a. Go to [https://app.run.ai/](https://app.run.ai/).
   b. Go to the Overview dashboard.
   c. Verify that the number of GPUs on the top right reflects the expected number of GPUs and the GPU nodes are all in the list of servers. For more information about Run:AI deployment, see [installing Run:AI](#).
Run:AI Dashboards and Views

After installing Run:AI on your Kubernetes cluster and configuring the containers correctly, you see the following dashboards and views on https://app.run.ai in your browser, as shown in the following figure.

There are 16 total GPUs in the cluster provided by two DGX-1 nodes. You can see the number of nodes, the total available GPUs, the allocated GPUs that are assigned with workloads, the total number of running jobs, pending jobs, and idle allocated GPUs. On the right side, the bar diagram shows GPUs per Project, which summarizes how different teams are using the cluster resource. In the middle is the list of currently running jobs with job details, including job name, project, user, job type, the node each job is running on, the number of GPU(s) allocated for that job, the current run time of the job, job progress in percentage, and the GPU utilization for that job. Note that the cluster is under-utilized (GPU utilization at 23%) because there are only three running jobs submitted by a single team (team-a).

In the following section, we show how to create multiple teams in the Projects tab and allocate GPUs for each team to maximize cluster usage and manage resources when there are many users per cluster. The test scenarios mimic enterprise environments in which memory and GPU resources are shared among training, inferencing, and interactive workloads.

Creating Projects for Data Science Teams and Allocating GPUs

Researchers can submit workloads through the Run:AI CLI, Kubeflow, or similar processes. To streamline resource allocation and create prioritization, Run:AI introduces the concept of Projects. Projects are quota entities that associate a project name with GPU allocation and preferences. It is a simple and convenient way to manage multiple data science teams.
A researcher submitting a workload must associate a project with a workload request. The Run:AI scheduler compares the request against the current allocations and the project and determines whether the workload can be allocated resources or whether it should remain in a pending state.

As a system administrator, you can set the following parameters in the Run:AI Projects tab:

- **Model projects.** Set a project per user, set a project per team of users, and set a project per a real organizational project.
- **Project quotas.** Each project is associated with a quota of GPUs that can be allocated for this project at the same time. This is a guaranteed quota in the sense that researchers using this project are guaranteed to get this number of GPUs no matter what the status in the cluster is. As a rule, the sum of the project allocation should be equal to the number of GPUs in the cluster. Beyond that, a user of this project can receive an over-quota. As long as GPUs are unused, a researcher using this project can get more GPUs. We demonstrate over-quota testing scenarios and fairness considerations in Achieving High Cluster Utilization with Over-Quota GPU Allocation, Basic Resource Allocation Fairness, and Over-Quota Fairness.
- Create a new project, update an existing project, and delete an existing project.
- **Limit jobs to run on specific node groups.** You can assign specific projects to run only on specific nodes. This is useful when the project team needs specialized hardware, for example, with enough memory. Alternatively, a project team might be the owner of specific hardware that was acquired with a specialized budget, or when you might need to direct build or interactive workloads to work on weaker hardware and direct longer training or unattended workloads to faster nodes. For commands to group nodes and set affinity for a specific project, see the Run:AI Documentation.
- **Limit the duration of interactive jobs.** Researchers frequently forget to close interactive jobs. This might lead to a waste of resources. Some organizations prefer to limit the duration of interactive jobs and close them automatically.

The following figure shows the Projects view with four teams created. Each team is assigned a different number of GPUs to account for different workloads, with the total number of GPUs equal to that of the total available GPUs in a cluster consisting of two DGX-1s.

---

**Submitting Jobs in Run:AI CLI**

This section provides the detail on basic Run:AI commands that you can use to run any Kubernetes job. It is divided into three parts according to workload type. AI/ML/DL workloads can be divided into two generic types:

- **Unattended training sessions.** With these types of workloads, the data scientist prepares a self-running workload and sends it for execution. During the execution, the customer can examine the results. This type
of workload is often used in production or when model development is at a stage where no human intervention is required.

- **Interactive build sessions.** With these types of workloads, the data scientist opens an interactive session with Bash, Jupyter Notebook, remote PyCharm, or similar IDEs and accesses GPU resources directly. We include a third scenario for running interactive workloads with connected ports to reveal an internal port to the container user.

### Unattended Training Workloads

After setting up projects and allocating GPU(s), you can run any Kubernetes workload using the following command at the command line:

```
$ runai project set team-a runai submit hyper1 -i gcr.io/run-ai-demo/quickstart -g 1
```

This command starts an unattended training job for team-a with an allocation of a single GPU. The job is based on a sample docker image, *gcr.io/run-ai-demo/quickstart*. We named the job **hyper1**. You can then monitor the job’s progress by running the following command:

```
$ runai list
```

The following figure shows the result of the `runai list` command. Typical statuses you might see include the following:

- **ContainerCreating.** The docker container is being downloaded from the cloud repository.
- **Pending.** The job is waiting to be scheduled.
- **Running.** The job is running.

To get an additional status on your job, run the following command:

```
$ runai get hyper1
```

To view the logs of the job, run the `runai logs <job-name>` command:

```
$ runai logs hyper1
```

In this example, you should see the log of a running DL session, including the current training epoch, ETA, loss function value, accuracy, and time elapsed for each step.

You can view the cluster status on the Run:AI UI at [https://app.run.ai/](https://app.run.ai/). Under Dashboards > Overview, you can monitor GPU utilization.
To stop this workload, run the following command:

```bash
$ runai delete hyper1
```

This command stops the training workload. You can verify this action by running `runai list` again. For more detail, see launching unattended training workloads.

**Interactive Build Workloads**

After setting up projects and allocating GPU(s) you can run an interactive build workload using the following command at the command line:

```bash
$ runai submit build1 -i python -g 1 --interactive --command sleep --args infinity
```

The job is based on a sample docker image python. We named the job build1.

The `--interactive` flag means that the job does not have a start or end. It is the researcher’s responsibility to close the job. The administrator can define a time limit for interactive jobs after which they are terminated by the system.

The `--g 1` flag allocates a single GPU to this job. The command and argument provided is `--command sleep --args infinity`. You must provide a command, or the container starts and then exits immediately.

The following commands work similarly to the commands described in Unattended Training Workloads:

- `runai list`: Shows the name, status, age, node, image, project, user, and GPUs for jobs.
- `runai get build1`: Displays additional status on the job build1.
- `runai delete build1`: Stops the interactive workload build1. To get a bash shell to the container, the following command:

```bash
$ runai bash build1
```

This provides a direct shell into the computer. Data scientists can then develop or finetune their models within the container.

You can view the cluster status on the Run:AI UI at `https://app.run.ai`. For more detail, see starting and using interactive build workloads.

**Interactive Workloads with Connected Ports**

As an extension of interactive build workloads, you can reveal internal ports to the container user when starting a container with the Run:AI CLI. This is useful for cloud environments, working with Jupyter Notebooks, or connecting to other microservices. Ingress allows access to Kubernetes services from outside the Kubernetes cluster. You can configure access by creating a collection of rules that define which inbound connections reach which services.
For better management of external access to the services in a cluster, we suggest that cluster administrators install Ingress and configure LoadBalancer.

To use Ingress as a service type, run the following command to set the method type and the ports when submitting your workload:

```
$ runai submit test-ingress -i jupyter/base-notebook -g 1 \
   --interactive --service-type=ingress --port 8888 \
   --args="--NotebookApp.base_url=test-ingress" --command=start-notebook.sh
```

After the container starts successfully, execute `runai list` to see the SERVICE URL(S) with which to access the Jupyter Notebook. The URL is composed of the ingress endpoint, the job name, and the port. For example, see `https://10.255.174.13/test-ingress-8888`.

For more details, see launching an interactive build workload with connected ports.

Achieving High Cluster Utilization

In this section, we emulate a realistic scenario in which four data science teams each submit their own workloads to demonstrate the Run:AI orchestration solution that achieves high cluster utilization while maintaining prioritization and balancing GPU resources. We start by using the ResNet-50 benchmark described in the section ResNet-50 with ImageNet Dataset Benchmark Summary:

```
$ runai submit netapp1 -i netapp/tensorflow-tf1-py3:20.01.0 --local-image \
   --large-shm -v /mnt:/mnt -v /tmp:/tmp --command python --args \
   "netapp/scripts/run.py" --args "--dataset_dir=/mnt/mount_0/dataset/imagenet/imagenet_original/" --args "--num_mounts=2" --args "--dgx_version=dgx1" --args "--num_devices=1" -g 1
```

We ran the same ResNet-50 benchmark as in NVA-1121. We used the flag `--local-image` for containers not residing in the public docker repository. We mounted the directories `/mnt` and `/tmp` on the host DGX-1 node to `/mnt` and `/tmp` to the container, respectively. The dataset is at NetApp AFFA800 with the `dataset_dir` argument pointing to the directory. Both `--num_devices=1` and `-g 1` mean that we allocate one GPU for this job. The former is an argument for the `run.py` script, while the latter is a flag for the `runai submit` command.

The following figure shows a system overview dashboard with 97% GPU utilization and all sixteen available GPUs allocated. You can easily see how many GPUs are allocated for each team in the GPUs/Project bar chart. The Running Jobs pane shows the current running job names, project, user, type, node, GPUs consumed, run time, progress, and utilization details. A list of workloads in queue with their wait time is shown in Pending Jobs. Finally, the Nodes box offers GPU numbers and utilization for individual DGX-1 nodes in the cluster.
Fractional GPU Allocation for Less Demanding or Interactive Workloads

When researchers and developers are working on their models, whether in the development, hyperparameter tuning, or debugging stages, such workloads usually require fewer computational resources. It is therefore more efficient to provision fractional GPU and memory such that the same GPU can simultaneously be allocated to other workloads. Run:AI’s orchestration solution provides a fractional GPU sharing system for containerized workloads on Kubernetes. The system supports workloads running CUDA programs and is especially suited for lightweight AI tasks such as inference and model building. The fractional GPU system transparently gives data science and AI engineering teams the ability to run multiple workloads simultaneously on a single GPU. This enables companies to run more workloads, such as computer vision, voice recognition, and natural language processing on the same hardware, thus lowering costs.

Run:AI’s fractional GPU system effectively creates virtualized logical GPUs with their own memory and computing space that containers can use and access as if they were self-contained processors. This enables several workloads to run in containers side-by-side on the same GPU without interfering with each other. The solution is transparent, simple, and portable and it requires no changes to the containers themselves.

A typical usecase could see two to eight jobs running on the same GPU, meaning that you could do eight times the work with the same hardware.

For the job frac05 belonging to project team-d in the following figure, we can see that the number of GPUs allocated was 0.50. This is further verified by the nvidia-smi command, which shows that the GPU memory available to the container was 16,255MB: half of the 32GB per V100 GPU in the DGX-1 node.
Achieving High Cluster Utilization with Over-Quota GPU Allocation

In this section and in the sections Basic Resource Allocation Fairness, and Over-Quota Fairness, we have devised advanced testing scenarios to demonstrate the Run:AI orchestration capabilities for complex workload management, automatic preemptive scheduling, and over-quota GPU provisioning. We did this to achieve high cluster-resource usage and optimize enterprise-level data science team productivity in an ONTAP AI environment.

For these three sections, set the following projects and quotas:

<table>
<thead>
<tr>
<th>Project</th>
<th>Quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>team-a</td>
<td>4</td>
</tr>
<tr>
<td>team-b</td>
<td>2</td>
</tr>
<tr>
<td>team-c</td>
<td>2</td>
</tr>
<tr>
<td>team-d</td>
<td>8</td>
</tr>
</tbody>
</table>

In addition, we use the following containers for these three sections:

- **Jupyter Notebook**: jupyter/base-notebook
- **Run:AI quickstart**: gcr.io/run-ai-demo/quickstart

We set the following goals for this test scenario:
• Show the simplicity of resource provisioning and how resources are abstracted from users
• Show how users can easily provision fractions of a GPU and integer number of GPUs
• Show how the system eliminates compute bottlenecks by allowing teams or users to go over their resource quota if there are free GPUs in the cluster
• Show how data pipeline bottlenecks are eliminated by using the NetApp solution when running compute-intensive jobs, such as the NetApp container
• Show how multiple types of containers are running using the system
  ◦ Jupyter Notebook
  ◦ Run:AI container
• Show high utilization when the cluster is full

For details on the actual command sequence executed during the testing, see Testing Details for Section 4.8.

When all 13 workloads are submitted, you can see a list of container names and GPUs allocated, as shown in the following figure. We have seven training and six interactive jobs, simulating four data science teams, each with their own models running or in development. For interactive jobs, individual developers are using Jupyter Notebooks to write or debug their code. Thus, it is suitable to provision GPU fractions without using too many cluster resources.

The results of this testing scenario show the following:

• The cluster should be full: 16/16 GPUs are used.
• High cluster utilization.
• More experiments than GPUs due to fractional allocation.
• team-d is not using all their quota; therefore, team-b and team-c can use additional GPUs for their experiments, leading to faster time to innovation.

Basic Resource Allocation Fairness

In this section, we show that, when team-d asks for more GPUs (they are under their quota), the system pauses the workloads of team-b and team-c and moves them into a pending state in a fair-share manner.

For details including job submissions, container images used, and command sequences executed, see the section Testing Details for Section 4.9.

The following figure shows the resulting cluster utilization, GPUs allocated per team, and pending jobs due to automatic load balancing and preemptive scheduling. We can observe that when the total number of GPUs
requested by all team workloads exceeds the total available GPUs in the cluster, Run:AI’s internal fairness algorithm pauses one job each for team-b and team-c because they have met their project quota. This provides overall high cluster utilization while data science teams still work under resource constraints set by an administrator.

The results of this testing scenario demonstrate the following:

- **Automatic load balancing.** The system automatically balances the quota of the GPUs, such that each team is now using their quota. The workloads that were paused belong to teams that were over their quota.

- **Fair share pause.** The system chooses to stop the workload of one team that was over their quota and then stop the workload of the other team. Run:AI has internal fairness algorithms.

**Over-Quota Fairness**

In this section, we expand the scenario in which multiple teams submit workloads and exceed their quota. In this way, we demonstrate how Run:AI’s fairness algorithm allocates cluster resources according to the ratio of preset quotas.

Goals for this test scenario:

- Show queuing mechanism when multiple teams are requesting GPUs over their quota.
- Show how the system distributes a fair share of the cluster between multiple teams that are over their quota according to the ratio between their quotas, so that the team with the larger quota gets a larger share of the spare capacity.

At the end of Basic Resource Allocation Fairness, there are two workloads queued: one for team-b and one for team-c. In this section, we queue additional workloads.
For details including job submissions, container images used, and command sequences executed, see Testing Details for section 4.10.

When all jobs are submitted according to the section Testing Details for section 4.10, the system dashboard shows that team-a, team-b, and team-c all have more GPUs than their preset quota. team-a occupies four more GPUs than its preset soft quota (four), whereas team-b and team-c each occupy two more GPUs than their soft quota (two). The ratio of over-quota GPUs allocated is equal to that of their preset quota. This is because the system used the preset quota as a reference of priority and provisioned accordingly when multiple teams request more GPUs, exceeding their quota. Such automatic load balancing provides fairness and prioritization when enterprise data science teams are actively engaged in AI model development and production.

The results of this testing scenario show the following:

- The system starts to de-queue the workloads of other teams.
- The order of the dequeuing is decided according to fairness algorithms, such that team-b and team-c get the same amount of over-quota GPUs (since they have a similar quota), and team-a gets a double amount of GPUs since their quota is two times higher than the quota of team-b and team-c.
- All the allocation is done automatically.

Therefore, the system should stabilize on the following states:

<table>
<thead>
<tr>
<th>Project</th>
<th>GPUs allocated</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>team-a</td>
<td>8/4</td>
<td>Four GPUs over the quota. Empty queue.</td>
</tr>
</tbody>
</table>
The following figure shows the GPU allocation per project over time in the Run:AI Analytics dashboard for the sections Achieving High Cluster Utilization with Over-Quota GPU Allocation, Basic Resource Allocation Fairness, and Over-Quota Fairness. Each line in the figure indicates the number of GPUs provisioned for a given data science team at any time. We can see that the system dynamically allocates GPUs according to workloads submitted. This allows teams to go over quota when there are available GPUs in the cluster, and then preempt jobs according to fairness, before finally reaching a stable state for all four teams.

<table>
<thead>
<tr>
<th>Project</th>
<th>GPUs allocated</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>team-b</td>
<td>4/2</td>
<td>Two GPUs over the quota. One workload queued.</td>
</tr>
<tr>
<td>team-c</td>
<td>4/2</td>
<td>Two GPUs over the quota. One workload queued.</td>
</tr>
<tr>
<td>team-d</td>
<td>0/8</td>
<td>Not using GPUs at all, no queued workloads.</td>
</tr>
</tbody>
</table>

Saving Data to a Trident-Provisioned PersistentVolume

NetApp Trident is a fully supported open source project designed to help you meet the sophisticated persistence demands of your containerized applications. You can read and write data to a Trident-provisioned Kubernetes PersistentVolume (PV) with the added benefit of data tiering, encryption, NetApp Snapshot technology, compliance, and high performance offered by NetApp ONTAP data management software.

Reusing PVCs in an Existing Namespace

For larger AI projects, it might be more efficient for different containers to read and write data to the same Kubernetes PV. To reuse a Kubernetes Persistent Volume Claim (PVC), the user must have already created a PVC. See the NetApp Trident documentation for details on creating a PVC. Here is an example of reusing an existing PVC:
Run the following command to see the status of job `pvc-test` for project `team-a`:

```
$ runai get pvc-test -p team-a
```

You should see the PV /tmp/pvc1mount mounted to `team-a` job `pvc-test`. In this way, multiple containers can read from the same volume, which is useful when there are multiple competing models in development or in production. Data scientists can build an ensemble of models and then combine prediction results by majority voting or other techniques.

Use the following to access the container shell:

```
$ runai bash pvc-test -p team-a
```

You can then check the mounted volume and access your data within the container.

This capability of reusing PVCs works with NetApp FlexVol volumes and NetApp ONTAP FlexGroup volumes, enabling data engineers more flexible and robust data management options to leverage your data fabric powered by NetApp.

**Conclusion**

NetApp and Run:AI have partnered in this technical report to demonstrate the unique capabilities of the NetApp ONTAP AI solution together with the Run:AI Platform for simplifying orchestration of AI workloads. The preceding steps provide a reference architecture to streamline the process of data pipelines and workload orchestration for deep learning. Customers looking to implement these solutions are encouraged to reach out to NetApp and Run:AI for more information.

**Testing Details for Section 4.8**

This section contains the testing details for the section *Achieving High Cluster Utilization with Over-Quota GPU Allocation*.

Submit jobs in the following order:

<table>
<thead>
<tr>
<th>Project</th>
<th>Image</th>
<th># GPUs</th>
<th>Total</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>team-a</td>
<td>Jupyter</td>
<td>1</td>
<td>1/4</td>
<td>–</td>
</tr>
<tr>
<td>team-a</td>
<td>NetApp</td>
<td>1</td>
<td>2/4</td>
<td>–</td>
</tr>
<tr>
<td>team-a</td>
<td>Run:AI</td>
<td>2</td>
<td>4/4</td>
<td>Using all their quota</td>
</tr>
<tr>
<td>team-b</td>
<td>Run:AI</td>
<td>0.6</td>
<td>0.6/2</td>
<td>Fractional GPU</td>
</tr>
<tr>
<td>Project</td>
<td>Image</td>
<td># GPUs</td>
<td>Total</td>
<td>Comment</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>--------</td>
<td>-------</td>
<td>-------------------</td>
</tr>
<tr>
<td>team-b</td>
<td>Run:AI</td>
<td>0.4</td>
<td>1/2</td>
<td>Fractional GPU</td>
</tr>
<tr>
<td>team-b</td>
<td>NetApp</td>
<td>1</td>
<td>2/2</td>
<td></td>
</tr>
<tr>
<td>team-b</td>
<td>NetApp</td>
<td>2</td>
<td>4/2</td>
<td>Two over quota</td>
</tr>
<tr>
<td>team-c</td>
<td>Run:AI</td>
<td>0.5</td>
<td>0.5/2</td>
<td>Fractional GPU</td>
</tr>
<tr>
<td>team-c</td>
<td>Run:AI</td>
<td>0.3</td>
<td>0.8/2</td>
<td>Fractional GPU</td>
</tr>
<tr>
<td>team-c</td>
<td>Run:AI</td>
<td>0.2</td>
<td>1/2</td>
<td>Fractional GPU</td>
</tr>
<tr>
<td>team-c</td>
<td>NetApp</td>
<td>2</td>
<td>3/2</td>
<td>One over quota</td>
</tr>
<tr>
<td>team-c</td>
<td>NetApp</td>
<td>1</td>
<td>4/2</td>
<td>Two over quota</td>
</tr>
<tr>
<td>team-d</td>
<td>NetApp</td>
<td>4</td>
<td>4/8</td>
<td>Using half of their quota</td>
</tr>
</tbody>
</table>

Command structure:

```
$ runai submit <job-name> -p <project-name> -g <#GPUs> -i <image-name>
```

Actual command sequence used in testing:

```
$ runai submit a-1-jupyter -i jupyter/base-notebook -g 1 --interactive --service-type=ingress --port 8888 --args="--NotebookApp.base_url=team-a-test-ingress" --command=start -notebook.sh -p team-a
$ runai submit a-1-g -i gcr.io/run-ai-demo/quickstart -g 1 -p team-a
$ runai submit a-2-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-a
$ runai submit b-1-g06 -i gcr.io/run-ai-demo/quickstart -g 0.6 --interactive -p team-b
$ runai submit b-2-g04 -i gcr.io/run-ai-demo/quickstart -g 0.4 --interactive -p team-b
$ runai submit b-3-g -i gcr.io/run-ai-demo/quickstart -g 1 -p team-b
$ runai submit b-4-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-b
$ runai submit c-1-g05 -i gcr.io/run-ai-demo/quickstart -g 0.5 --interactive -p team-c
$ runai submit c-2-g03 -i gcr.io/run-ai-demo/quickstart -g 0.3 --interactive -p team-c
$ runai submit c-3-g02 -i gcr.io/run-ai-demo/quickstart -g 0.2 --interactive -p team-c
$ runai submit c-4-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-c
$ runai submit c-5-g -i gcr.io/run-ai-demo/quickstart -g 1 -p team-c
$ runai submit d-1-gggg -i gcr.io/run-ai-demo/quickstart -g 4 -p team-d
```

At this point, you should have the following states:
### Testing Details for Section 4.9

This section contains testing details for the section **Basic Resource Allocation Fairness**.

Submit jobs in the following order:

<table>
<thead>
<tr>
<th>Project</th>
<th># GPUs</th>
<th>Total</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>team-d</td>
<td>2</td>
<td>6/8</td>
<td>Team-b/c workload pauses and moves to pending.</td>
</tr>
<tr>
<td>team-d</td>
<td>2</td>
<td>8/8</td>
<td>Other team (b/c) workloads pause and move to pending.</td>
</tr>
</tbody>
</table>

See the following executed command sequence:

```
$ runai submit d-2-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-d
runai submit d-3-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-d
```

At this point, you should have the following states:

<table>
<thead>
<tr>
<th>Project</th>
<th>GPUs Allocated</th>
<th>Workloads Queued</th>
</tr>
</thead>
<tbody>
<tr>
<td>team-a</td>
<td>4/4</td>
<td>None</td>
</tr>
<tr>
<td>team-b</td>
<td>2/2</td>
<td>None</td>
</tr>
<tr>
<td>team-c</td>
<td>2/2</td>
<td>None</td>
</tr>
<tr>
<td>team-d</td>
<td>8/8</td>
<td>None</td>
</tr>
</tbody>
</table>

See the section **Basic Resource Allocation Fairness** for a discussion on the proceeding testing scenario.

### Testing Details for Section 4.10

This section contains testing details for the section **Over-Quota Fairness**.

Submit jobs in the following order for **team-a**, **team-b**, and **team-c**.
<table>
<thead>
<tr>
<th>Project</th>
<th># GPUs</th>
<th>Total</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>team-a</td>
<td>2</td>
<td>4/4</td>
<td>1 workload queued</td>
</tr>
<tr>
<td>team-a</td>
<td>2</td>
<td>4/4</td>
<td>2 workloads queued</td>
</tr>
<tr>
<td>team-b</td>
<td>2</td>
<td>2/2</td>
<td>2 workloads queued</td>
</tr>
<tr>
<td>team-c</td>
<td>2</td>
<td>2/2</td>
<td>2 workloads queued</td>
</tr>
</tbody>
</table>

See the following executed command sequence:

```
$ runai submit a-3-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-a$
runai submit a-4-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-a$
runai submit b-5-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-b$
runai submit c-6-gg -i gcr.io/run-ai-demo/quickstart -g 2 -p team-c
```

At this point, you should have the following states:

<table>
<thead>
<tr>
<th>Project</th>
<th>GPUs Allocated</th>
<th>Workloads Queued</th>
</tr>
</thead>
<tbody>
<tr>
<td>team-a</td>
<td>4/4</td>
<td>Two workloads asking for GPUs two each</td>
</tr>
<tr>
<td>team-b</td>
<td>2/2</td>
<td>Two workloads asking for two GPUs each</td>
</tr>
<tr>
<td>team-c</td>
<td>2/2</td>
<td>Two workloads asking for two GPUs each</td>
</tr>
<tr>
<td>team-d</td>
<td>8/8</td>
<td>None</td>
</tr>
</tbody>
</table>

Next, delete all the workloads for team-d:

```
$ runai delete -p team-d d-1-gggg d-2-gg d-3-gg
```

See the section Over-Quota Fairness, for discussions on the proceeding testing scenario.

**Where to Find Additional Information**

To learn more about the information that is described in this document, see the following resources:

- NVIDIA DGX Systems
  - NVIDIA DGX-1 System
  - NVIDIA V100 Tensor Core GPU
  - NVIDIA NGC
TR-4799-DESIGN: NetApp ONTAP AI reference architecture for autonomous driving workloads

David Arnette and Sung-Han Lin, NetApp

The NVIDIA DGX family of systems is the world’s first integrated artificial intelligence (AI) platform that is purpose-built for enterprise AI. NetApp AFF storage systems deliver
extreme performance and industry-leading hybrid cloud data-management capabilities. NetApp and NVIDIA have partnered to create the NetApp ONTAP AI reference architecture to offer customers a turnkey solution for supporting AI and machine learning (ML) workloads with enterprise-class performance, reliability, and support.

**TR-4799-DESIGN: NetApp ONTAP AI reference architecture for autonomous driving workloads**

**TR-4811: NetApp ONTAP AI reference architecture for healthcare: Diagnostic imaging - Solution design**

Rick Huang, Sung-Han Lin, Sathish Thyagarajan, NetApp
Jacci Cenci, NVIDIA

This reference architecture offers guidelines for customers building artificial intelligence (AI) infrastructure using NVIDIA DGX-2 systems and NetApp AFF storage for healthcare use cases. It includes information about the high-level workflows used in the development of deep learning (DL) models for medical diagnostic imaging, validated test cases, and results. It also includes sizing recommendations for customer deployments.

**TR-4811: NetApp ONTAP AI reference architecture for healthcare: Diagnostic imaging - Solution design**

**TR-4807: NetApp ONTAP AI reference architecture for financial services workloads - Solution design**

Karthikeyan Nagalingam, Sung-Han Lin, NetApp
Jacci Cenci, NVIDIA

This reference architecture offers guidelines for customers who are building artificial intelligence infrastructure using NVIDIA DGX-1 systems and NetApp AFF storage for financial sector use cases. It includes information about the high-level workflows used in the development of deep learning models for financial services test cases and results. It also includes sizing recommendations for customer deployments.

**TR-4807: NetApp ONTAP AI reference architecture for financial services workloads - Solution design**

**Generative AI and NetApp Value**

Author: Sathish Thyagarajan, NetApp

**Abstract**

The demand for generative artificial intelligence (AI) is driving disruption across industries, enhancing business creativity and product innovation. Many organizations are using generative AI to build new product features, improve engineering productivity and prototype AI powered applications that deliver better results and consumer experiences. Generative AI such as Generative Pre-trained Transformers (GPT) use neural networks to create new content, as diverse as text, audio, and video. Given the extreme scale and massive datasets involved with large language models (LLMs), it is crucial to architect a robust AI infrastructure that takes advantage of the compelling data storage features of on-premises, hybrid and multicloud deployment options and reduce risks associated with data mobility, data protection and governance before companies can design AI solutions. This paper describes these considerations and the corresponding NetApp® AI capabilities.
that enable seamless data management and data movement across the AI data pipeline for training, retraining, fine-tuning, and inferencing generative AI models.

Executive Summary

Most recently after the launch of ChatGPT, a spin-off of GPT-3 in November 2022, new AI tools used to generate text, code, image, or even therapeutic proteins in response to user prompts have gained significant fame. This indicates users can make a request using natural language and AI will interpret and generate text, such as news articles or product descriptions that reflect user request or produce code, music, speech, visual effects, and 3D assets using algorithms trained on already existing data. As a result, phrases like Stable Diffusion, Hallucinations, Prompt Engineering and Value Alignment are rapidly emerging in the design of AI systems. These self-supervised or semi-supervised machine learning (ML) models are becoming widely available as pre-trained foundation models (FM) via cloud service providers and other AI firms/vendors, which are being adopted by various business establishments across industries for a wide range of downstream NLP (natural language processing) tasks. As asserted by research analyst firms like McKinsey – “Generative AI’s impact on productivity could add trillions of dollars in value to the global economy.” While companies are reimagining AI as thought partners to humans and FMs are broadening simultaneously to what businesses and institutions can do with generative AI, the opportunities to manage massive volumes of data will continue to grow. This document presents introductory information on generative AI and the design concepts in relation to NetApp capabilities that bring value to NetApp customers, both on-premises and hybrid or multicloud environments.

So, what’s in it for customers to use NetApp in their AI environments? NetApp helps organizations meet the complexities created by rapid data and cloud growth, multi-cloud management, and the adoption of next-generation technologies, such as AI. NetApp has combined various capabilities into intelligent data management software and storage infrastructure that have been well balanced with high-performance optimized for AI workloads. Generative AI solutions like LLMs need to read and process their source datasets from storage into memory numerous times to foster intelligence. NetApp has been a leader in data mobility, data governance and data security technologies across the edge-to-core-to-cloud ecosystem, serving enterprise customers build at-scale AI solutions. NetApp, with a strong network of partners has been helping chief data officers, AI engineers, enterprise architects and data scientists in the design of a free-flowing data pipeline for data preparation, data protection, and strategic data management responsibilities of AI model training and inferencing, optimizing the performance and scalability of the AI/ML lifecycle. NetApp data technologies and capabilities such as NetApp® ONTAP AI® for deep learning data pipeline, NetApp® SnapMirror® for transporting data seamlessly and efficiently between storage endpoints, and NetApp® FlexCache® for real-time rendering when the data flow shifts from batch to real-time and data engineering happens at prompt time, bring value to the deployment of real-time Generative AI models. As enterprises of all types embrace new AI tools, they face data challenges from the edge to the data center to the cloud that demand for scalable, responsible and explainable AI solutions. As the data authority on hybrid and multi cloud, NetApp is committed to building a network of partners and joint solutions that can help with all aspects of constructing a data pipeline and data lakes for generative AI model training (pre-training), fine-tuning, context-based inferencing and model decay monitoring of LLMs.

What is Generative AI?

Generative AI is changing how we create content, generate new design concepts, and explore novel compositions. It illustrates neural network frameworks like Generative Adversarial Network (GAN), Variational Autoencoders (VAE), and Generative Pre-Trained Transformers (GPT), which can generate new content like text, code, images, audio, video, and synthetic data. Transformer-based models like OpenAI’s Chat-GPT, Google’s Bard, Hugging Face’s BLOOM, and Meta’s LLaMA have emerged as the foundational technology underpinning many advances in large language models. Likewise, OpenAI’s Dall-E, Meta’s CM3leon, and Google’s Imagen are examples for text-to-image diffusion models which offer customers an unprecedented degree of photorealism to create new, complex images from scratch or edit existing images to generate high-quality context-aware images using dataset augmentation and text-to-image synthesis linking textual and visual semantics. Digital artists are starting to apply a combination of rendering technologies like NeRF (Neural
Radiance Field) with generative AI to convert static 2D images into immersive 3D scenes. In general, LLMs are broadly characterized by four parameters: (1) Size of the model (typically in billions of parameters); (2) Size of the training dataset; (3) Cost of training, and (4) Model performance after training. LLMs also fall mainly into three transformer architectures. (i) Encoder-only models. E.g. BERT (Google, 2018); (ii) Encoder-Decoder E.g. BART (Meta, 2020) and (iii) Decoder-only models. E.g. LLaMA (Meta, 2023), PaLM-E (Google, 2023).

Depending on the business requirement, irrespective of which architecture a company chooses the number of model parameters (N) and the number of tokens (D) in the training dataset generally determine the baseline cost of training (pre-training) or fine-tuning an LLM.

**Enterprise Use Cases and Downstream NLP Tasks**

Businesses across industries are uncovering more and more potential for AI to extract and produce new forms of value from existing data for business operations, sales, marketing, and legal services. According to IDC (International Data Corporation) market intelligence on global generative AI use cases and investments, knowledge management in software development and product design is to be the most impacted, followed by storyline creation for marketing and code generation for developers. In healthcare, clinical research organizations are breaking new ground in medicine. Pretrained models like ProteinBERT incorporate Gene Ontology (GO) annotations to rapidly design protein structures for medical drugs, representing a significant milestone in drug discovery, bioinformatics, and molecular biology. Biotech firms have initiated human trials for generative AI-discovered medicine, that aims to treat diseases like pulmonary fibrosis (IPF), a lung disease that causes irreversible scarring of lung tissue.

Figure 1: Use cases driving Generative AI

<table>
<thead>
<tr>
<th>Chatbots</th>
<th>Drug discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text generation</td>
<td>Genome model expression</td>
</tr>
<tr>
<td>Image generation</td>
<td>Classification</td>
</tr>
<tr>
<td>Code generation</td>
<td>Speech-to-Text</td>
</tr>
</tbody>
</table>

Increases in automation adoption driven by generative AI is also changing the supply & demand of work activities for many occupations. As per McKinsey the US labor market (diagram below) has gone through a rapid transition, which may only continue when factoring in the impact of AI.

Source: McKinsey & Company
Role of storage in generative AI

LLMs rely largely on deep learning, GPUs, and compute. However, when GPU buffer fills up, the data needs to be written quickly to storage. While some AI models are small enough to execute in memory, LLMs require high IOPS and high throughput storage to provide fast access to large datasets, especially if it involves billions of tokens or millions of images. For a typical GPU memory requirement of an LLM, the memory needed to train a model with 1 billion parameters could go up to 80GB @32-bit full precision. In which case, Meta’s LLaMA 2, a family of LLMs ranging in scale from 7 billion to 70 billion parameters, may require 70x80, approx. 5600GB or 5.6TB of GPU RAM. Furthermore, the amount of memory you need is directly proportional to the maximum number of tokens you want to generate. For example, if you want to generate outputs of up to 512 tokens (about 380 words), you need “512MB”. It may seem inconsequential – but, if you want to run bigger batches it starts to add up. Therefore, making it very expensive for organizations training or fine-tuning LLMs in memory, thus making storage a cornerstone for generative AI.

Three primary approaches to LLMs

For most businesses, based on current trends, the approach to deploying LLMs can be condensed into 3 basic scenarios. As described in a recent “Harvard Business Review” article: (1) Training (pre-training) an LLM from
scratch – costly and requires expert AI/ML skills; (2) Fine-tuning a foundation model with enterprise data – complex, yet feasible; (3) Using retrieval-augmented generation (RAG) to query document repositories, APIs and vector databases that contain company data. Each of these has tradeoffs between the effort, iteration speed, cost-efficiency and model accuracy in their implementations, used to solving different types of problems (diagram below).

Figure 3: Problem Types

Foundation Models

A foundation model (FM) also known as base model is a large AI model (LLM) trained on vast quantities of unlabeled data, using self-supervision at scale, generally adapted for a wide range of downstream NLP tasks. Since the training data is not labelled by humans, the model emerges rather than being explicitly encoded. This means the model can generate stories or a narrative of its own without being explicitly programmed to do so. Hence an important characteristic of FM is homogenization, which means the same method is used in many domains. However, with personalization and fine-tuning techniques, FMs integrated into products appearing these days are not only good at generating text, text-to-images, and text-to-code, but also for explaining domain specific tasks or debugging code. For instance, FMs like OpenAI’s Codex or Meta’s Code Llama can generate code in multiple programming languages based on natural language descriptions of a programming task. These models are proficient in over a dozen programming languages including Python, C#, JavaScript, Perl, Ruby, and SQL. They understand the user’s intent and generate specific code that accomplishes the desired task useful for software development, code optimization, and automation of programming tasks.

Fine-tuning, domain-specificity, and retraining

One of the common practices with LLM deployment following data preparation and data pre-processing is to select a pre-trained model that has been trained on a large and diverse dataset. In the context of fine-tuning this can be an open-source large language model such as "Meta’s Llama 2" trained on 70 billion parameters and 2 trillion tokens. Once the pre-trained model is selected, the next step is to fine-tune it on the domain-specific data. This involves adjusting the model's parameters and training it on the new data to adapt to a specific domain and task. For example, BloombergGPT, a proprietary LLM trained on a wide range of financial data serving the financial industry. Domain-specific models designed and trained for a specific task generally have higher accuracy and performance within their scope, but low transferability across other tasks or domains. When business environment and data change over a period, the prediction accuracy of the FM could begin to decline when compared to their performance during testing. This is when retraining or fine-tuning the model becomes crucial. Model retraining in traditional AI/ML refers to updating a deployed ML model with new data, generally performed to eliminate two types of drifts that occur. (1) Concept drift – when the link between the input variables and the target variables changes over time, since the description of what we want to predict changes, the model can produce inaccurate predictions. (2) Data drift – occurs when the characteristics of the input data change, like changes in customer habits or behavior over time and therefore the model’s inability to respond to such changes. In a similar fashion, retraining applies to FMs/LLMs, however it can be a lot costlier
(in $millions), therefore not something most organizations might consider. It is under active research, still emerging in the realm of LLMOps. So instead of re-training, when model decay occurs in fine-tuned FMs, businesses may opt for fine-tuning again (lot cheaper) with a newer dataset. For a cost perspective, listed below is an example of a model-price table from Azure-OpenAI Services. For each task category, customers can fine-tune and evaluate models on specific datasets.

Source: Microsoft Azure

<table>
<thead>
<tr>
<th>Model</th>
<th>Per 1000 token</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text-Ada</td>
<td>$0.0001</td>
</tr>
<tr>
<td>GPT-3.5 Turbo</td>
<td>$0.003</td>
</tr>
<tr>
<td>GPT-4</td>
<td>$0.06</td>
</tr>
<tr>
<td>Text-Davinci</td>
<td>$0.02</td>
</tr>
<tr>
<td><strong>Model</strong></td>
<td><strong>Per 100 images</strong></td>
</tr>
<tr>
<td>Dall-E</td>
<td>$2</td>
</tr>
</tbody>
</table>

Prompt engineering and Inferencing

Prompt engineering refers to the effective methods of how to communicate with LLMs to perform desired tasks without updating the model weights. As important as AI model training and fine-tuning is to NLP applications, inferencing is equally important, where the trained models respond to user prompts. The system requirements for inferencing are generally much more on the read performance of the AI storage system that feeds data from LLMs to the GPUs as it needs to be able to apply billions of stored model parameters to produce the best response.

LLMOps, Model Monitoring and Vectorstores

Like traditional Machine Learning Ops (MLOps), Large Language Model Operations (LLMOps) also require the collaboration of data scientists and DevOps engineers with tools and best practices for the management of LLMs in production environments. However, the workflow and tech stack for LLMs could vary in some ways. For instance, LLM pipelines built using frameworks like LangChain string together multiple LLM API calls to external embedding endpoints such as vectorstores or vector databases. The use of an embedding endpoint and vectorstore for downstream connectors (like to a vector database) represents a significant development in how data is stored and accessed. As opposed to traditional ML models that are developed from scratch, LLMs often rely on transfer learning since these models start with FMs that are fine-tuned with new data to improve performance in a more specific domain. Therefore, it is crucial LLMOps deliver the capabilities of risk management and model decay monitoring.

Risks and Ethics in the age of Generative AI

"ChatGPT – It's slick but still spews nonsense."– MIT Tech Review. Garbage in–garbage out, has always been the challenging case with computing. The only difference with generative AI is that it excels at making the garbage highly credible, leading to inaccurate outcomes. LLMs are prone to invent facts to fit the narrative it's building. Therefore, companies that see generative AI as a great opportunity to lower their costs with AI equivalents need to efficiently detect deep fakes, reduce biases, and lower risks to keep the systems honest and ethical. A free-flowing data pipeline with a robust AI infrastructure that supports data mobility, data quality, data governance and data protection via end-to-end encryption and AI guardrails is eminent in the design of
responsible and explainable generative AI models.

**Customer scenario and NetApp**

Figure 3: Machine Learning/Large Language Model Workflow

- Data preparation
- Model monitoring
- Model Training
- Model deployment/Inferencing
- Model evaluation

---

**Are we training or fine-tuning?** The question of whether to (a) train an LLM model from scratch, fine-tune a pre-trained FM, or use RAG to retrieve data from document repositories outside a foundation model and augment prompts, and (b) either by leveraging open-source LLMs (E.g., Llama 2) or proprietary FMs (E.g., ChatGPT, Bard, AWS Bedrock) is a strategic decision for organizations. Each approach has a tradeoff between cost-efficiency, data gravity, operations, model accuracy and management of LLMs.

NetApp as a company embraces AI internally in its work culture and in its approach to product design and engineering efforts. For instance, NetApp’s autonomous ransomware protection is built using AI and machine learning. It provides early detection of file system anomalies to help identify threats before they impact operations. Second, NetApp uses predictive AI for its business operations like sales and inventory forecasting and chatbots to assist customers in call center product support services, tech specs, warranty, service manuals, and more. Third, NetApp brings customer value to the AI data pipeline and ML/LLM workflow via products and solutions serving customers building predictive AI solutions such as demand forecasting, medical imaging, sentiment analysis, and generative AI solutions like GANs for industrial images anomaly detection and product quality in manufacturing sector and anti-money laundering and fraud detection in banking & financial services with NetApp products and capabilities like NetApp® ONTAP AI®, NetApp® SnapMirror®, and NetApp® FlexCache®.

**NetApp capabilities**

The movement and management of data in generative AI applications such as chatbot, code generation, image generation or genome model expression can span across the edge, private data center, and hybrid multicloud ecosystem. For instance, a real-time AI-bot helping a passenger upgrade his or her airline ticket to business class from an end-user app exposed via APIs of pre-trained models such as ChatGPT cannot achieve that task by itself since the passenger information is not publicly available on the internet. The API
requires access to the passenger’s personal info and ticket info from the airline carrier which may exist in a hybrid or multicloud ecosystem. A similar scenario might apply to scientists sharing a drug molecule and patient data via an end-user application that uses LLMs to accomplish clinical trials across drug discovery involving one-to-many bio-medical research institutions. Sensitive data that gets passed to FMs or LLMs may include PII, financial information, health information, biometric data, location data, communications data, online behavior, and legal information. In such an event of real-time rendering, prompt execution and edge inferencing there is data movement from end user app to storage endpoints via open source or proprietary LLM models to a data center on premises or public cloud platforms. In all such scenarios, data mobility and data protection are crucial for the AI operations involving LLMs which rely on large training datasets and movement of such data.

Figure 4: Generative AI - LLM Data Pipeline

NetApp’s portfolio of storage infrastructure, data and cloud services is powered by intelligent data management software.

**Data Preparation**: The first pillar of the LLM tech stack is largely untouched from the older traditional ML stack. Data preprocessing in AI pipeline is necessary to normalize and cleanse the data before training or fine-tuning. This step includes connectors to ingest data wherever it may reside in the form of an Amazon S3 tier or in on-premises storage systems such as a file store or an object store like NetApp StorageGRID.

**NetApp® ONTAP** is the foundational technology that underpins NetApp’s critical storage solutions in the data center and the cloud. ONTAP includes various data management and protection features and capabilities, including automatic ransomware protection against cyber-attacks, built-in data transport features, and storage efficiency capabilities for a range of architectures from on-premises, hybrid, multiclouds in NAS, SAN, object, and software defined storage (SDS) situations of LLM deployments.

**NetApp® ONTAP AI®** for deep learning model training. NetApp® ONTAP® supports NVIDIA GPU Direct Storage™ with the use of NFS over RDMA for NetApp customers with ONTAP storage cluster and NVIDIA DGX compute nodes. It offers a cost-efficient performance to read and process source datasets from storage into memory numerous times to foster intelligence, enabling organizations with training, fine-tuning, and scaling access to LLMs.
NetApp® FlexCache® is a remote caching capability that simplifies file distribution and caches only the actively read data. This can be useful for LLM training, re-training, and fine-tuning, bringing value to customers with business requirements like real-time rendering and LLM inferencing.

NetApp® SnapMirror is an ONTAP feature that replicates volume snapshots between any two ONTAP systems. This feature optimally transfers data at the edge to your on-premises data center or to the cloud. SnapMirror can be used for moving data securely and efficiently between on-premises and hyperscaler clouds, when customers want to develop generative AI in clouds with RAG containing enterprise data. It efficiently transfers only changes, saving bandwidth and speeding replication, thus bringing essential data mobility features during the operations of training, re-training, and fine-tuning of FMs or LLMs.

NetApp® SnapLock brings immutable disk capability on ONTAP-based storage systems for dataset versioning. The microcore architecture is designed to protect customer data with FPolicy™ Zero Trust engine. NetApp ensures customer data is available by resisting denial-of-service (DoS) attacks when an attacker interacts with an LLM in a particularly resource-consuming way.

NetApp® Cloud Data Sense helps identify, map, and classify personal information present in enterprise datasets, enact policies, meet privacy requirements on premises or in the cloud, help improve security posture and comply with regulations.

NetApp® BlueXP™ classification, powered by Cloud Data Sense. Customers can automatically scan, analyze, categorize, and act on data across data estate, detect security risks, optimize storage, and accelerate cloud deployments. It combines storage and data services via its unified control plane. Customers can use GPU instances for computation, and hybrid multicloud environments for cold storage tiering and for archives and backups.

NetApp File-Object Duality. NetApp ONTAP enables dual-protocol access for NFS and S3. With this solution, customers can access NFS data from Amazon AWS SageMaker notebooks via S3 buckets from NetApp Cloud Volumes ONTAP. This offers flexibility to customers who need easy access to heterogenous data sources with the ability to share data from both NFS and S3. For e.g., fine-tuning FMs like Meta's Llama 2 text-generation models on SageMaker with access to file-object buckets.

NetApp® Cloud Sync service offers a simple and secure way to migrate data to any target, in the cloud or on-premises. Cloud Sync seamlessly transfers and synchronizes data between on-premises or cloud storage, NAS, and object stores.

NetApp XCP is a client software that enables fast and reliable any-to-NetApp and NetApp-to-NetApp data migrations. XCP also provides the capability of moving bulk data efficiently from Hadoop HDFS file systems into ONTAP NFS, S3 or StorageGRID and XCP file analytics provides visibility into the file system.

NetApp® DataOps Toolkit is a Python library that makes it simple for data scientists, DevOps, and data engineers to perform various data management tasks, such as near-instantaneously provisioning, cloning, or snapshotting a data volume or JupyterLab workspace that are backed by high-performance scale-out NetApp storage.

NetApp’s product security. LLMs may inadvertently reveal confidential data in their responses, thus a concern to CISOs who study the vulnerabilities associated with AI applications leveraging LLMs. As outlined by OWASP (Open Worldwide Application Security Project), security issues such as data poisoning, data leakage, denial of service and prompt injections within LLMs can impact businesses from data exposure to unauthorized access serving attackers. Data storage requirements should include integrity checks and immutable snapshots for structured, semi-structured, and unstructured data. NetApp Snapshots and SnapLock are being used for dataset versioning. It brings strict role-based access control (RBAC), as well as secure protocols, and industry standard encryption for securing both data at rest and in transit. Cloud Insights and Cloud Data Sense together offer capabilities to help you forensically identify the source of the threat and prioritize which data to restore.
ONTAP AI with DGX BasePOD

NetApp® ONTAP® AI reference architecture with NVIDIA DGX BasePOD is a scalable architecture for machine learning (ML) and artificial intelligence (AI) workloads. For the critical training phase of LLMs, data is typically copied from the data storage into the training cluster at regular intervals. The servers that are used in this phase use GPUs to parallelize computations, creating a tremendous appetite for data. Meeting the raw I/O bandwidth needs is crucial for maintaining high GPU utilization.

ONTAP AI with NVIDIA AI Enterprise

NVIDIA AI Enterprise is an end-to-end, cloud-native suite of AI and data analytics software that is optimized, certified, and supported by NVIDIA to run on VMware vSphere with NVIDIA-Certified Systems. This software facilitates the simple and rapid deployment, management, and scaling of AI workloads in the modern hybrid cloud environment. NVIDIA AI Enterprise, powered by NetApp and VMware, delivers enterprise-class AI workload and data management in a simplified, familiar package.

1P Cloud Platforms

Fully managed cloud storage offerings are available natively on Microsoft Azure as Azure NetApp Files (ANF), on AWS as Amazon FSx for NetApp ONTAP (FSxN), and on Google as Google Cloud NetApp Volumes (GNCV). 1P is a managed, high-performance file system that enables customers to run highly available AI workloads with improved data security in public clouds, for fine-tuning LLMs/FMs with cloud native ML platforms like AWS SageMaker, Azure-OpenAI Services, and Google’s Vertex AI.

NetApp Partner Solution Suite

In addition to its core data products, technologies and capabilities, NetApp also collaborates closely with a robust network of AI partners to bring added value to customers.

NVIDIA Guardrails in AI systems serve as safeguards to ensure the ethical and responsible use of AI technologies. AI developers can choose to define the behavior of LLM-powered applications on specific topics and prevent them from engaging in discussions on unwanted topics. Guardrails, an open-source toolkit, provides the ability to connect an LLM to other services, seamlessly and securely for building trustworthy, safe, and secure LLM conversational systems.

Domino Data Lab provides versatile, enterprise-grade tools for building and productizing Generative AI - fast, safe, and economical, wherever you are in your AI journey. With Domino’s Enterprise MLOps Platform, data scientists can use preferred tools and all their data, train and deploy models easily anywhere and manage risk and cost effectively - all from one control center.

Modzy for Edge AI. NetApp® and Modzy have partnered together to deliver AI at scale to any type of data, including imagery, audio, text, and tables. Modzy is an MLOps platform for deploying, integrating, and running AI models, offers data scientists the capabilities of model monitoring, drift detection and explainability, with an integrated solution for seamless LLM inference.

Run:AI and NetApp have partnered to demonstrate the unique capabilities of the NetApp ONTAP AI solution with the Run:AI cluster management platform for simplifying orchestration of AI workloads. It automatically splits and joins GPU resources, designed to scale your data processing pipelines to hundreds of machines with built-in integration frameworks for Spark, Ray, Dask, and Rapids.

Conclusion

Generative AI can produce effective results only when the model is trained on reams of quality data. While LLMs have achieved remarkable milestones, it is critical to recognize its limitations, design challenges and risks associated with data mobility and data quality. LLMs rely on large and disparate training datasets from
heterogenous data sources. Inaccurate outcomes or biased results generated by the models can put both businesses and consumers in jeopardy. These risks can correspond to constraints for LLMs emerging potentially from data management challenges associated with data quality, data security, and data mobility. NetApp helps organizations meet the complexities created by rapid data growth, data mobility, multi-cloud management, and the adoption of AI. At scale AI infrastructure and efficient data management is crucial to defining the success of AI applications like generative AI. It is critical customers cover all the deployment scenarios without compromising on the ability to expand as enterprises need to while maintaining cost-efficiency, data governance and ethical AI practices in control. NetApp is constantly working to help customers simplify and accelerate their AI deployments.

**TR-4785: AI Deployment with NetApp E-Series and BeeGFS**

Nagalakshmi Raju, Daniel Landes, Nathan Swartz, Amine Bennani, NetApp

Artificial intelligence (AI), machine learning (ML), and deep learning (DL) applications involve large datasets and high computations. To run these workloads successfully, you need an agile infrastructure that allows you to scale out both storage and compute nodes seamlessly. This report includes the steps for running an AI training model in a distributed mode, which allows seamless scale-out of compute and storage nodes. The report also includes various performance metrics to show how a solution combining NetApp E-Series storage with the BeeGFS parallel file system provides a flexible, cost-effective, and simple solution for AI workloads.

**NVA-1150-DESIGN: Quantum StorNext with NetApp E-Series systems design guide**

Ryan Rodine, NetApp

This document provides details on how to design a StorNext parallel file system solution with NetApp E-Series storage systems. This solution covers the NetApp EF280 all-flash array, the NetApp EF300 all-flash NVMe array, the EF600 all-flash NVMe array, and the NetApp E5760 hybrid system. It offers performance characterization based on Frametest benchmarking, a tool that is widely used for testing in the media and entertainment industry.

**NVA-1150-DEPLOY: Quantum StorNext with NetApp E-Series systems deployment guide**

Ryan Rodine, NetApp

This document provides details on how to deploy a StorNext parallel file system solution with NetApp E-Series storage systems. This solution covers the NetApp EF280 all-flash array, the NetApp EF300 all-flash NVMe array, the NetApp EF600 all-flash NVMe array, and the NetApp E5760 hybrid system. It offers performance characterization based on Frametest benchmarking, a tool that is widely used for testing in the media and entertainment industry.
NetApp Container Solutions

NVA-1165: Anthos with NetApp

Banu Sundhar and Suresh Thoppay, NetApp

This reference document provides deployment validation of the Anthos with NetApp solution by NetApp and our engineering partners when it is deployed in multiple data-center environments. It also details storage integration with NetApp storage systems by using the Astra Trident storage orchestrator for the management of persistent storage. Lastly, we explore and document a number of solution validations and real-world use cases.

Use cases

The Anthos with NetApp solution is architected to deliver exceptional value for customers with the following use cases:

• Easy to deploy and manage Anthos environment deployed using the provided bmctl tool on bare metal or the gkectl tool on VMware vSphere.

• Combined power of enterprise container and virtualized workloads with Anthos deployed virtually on vSphere or on bare metal with kubevirt.

• Real-world configuration and use cases highlighting Anthos features when used with NetApp storage and Astra Trident, the open-source storage orchestrator for Kubernetes.

Business value

Enterprises are increasingly adopting DevOps practices to create new products, shorten release cycles, and rapidly add new features. Because of their innate agile nature, containers and microservices play a crucial role in supporting DevOps practices. However, practicing DevOps at a production scale in an enterprise environment presents its own challenges and imposes certain requirements on the underlying infrastructure, such as the following:

• High availability at all layers in the stack
• Ease of deployment procedures
• Non-disruptive operations and upgrades
• API-driven and programmable infrastructure to keep up with microservices agility
• Multitenancy with performance guarantees
• The ability to run virtualized and containerized workloads simultaneously
• The ability to scale infrastructure independently based on workload demands

The Anthos with NetApp solution acknowledges these challenges and presents a solution that helps address each concern by implementing the fully automated deployment of Anthos on prem in the customer’s data center environment of choice.
Technology overview

The Anthos with NetApp solution is comprised of the following major components:

**Anthos On Prem**

Anthos On Prem is a fully supported enterprise Kubernetes platform that can be deployed in the VMware vSphere hypervisor, or on a bare metal infrastructure of your choosing.

For more information about Anthos, see the Anthos website located [here](#).

**NetApp storage systems**

NetApp has several storage systems perfect for enterprise data centers and hybrid cloud deployments. The NetApp portfolio includes NetApp ONTAP, Cloud Volumes ONTAP, Cloud Volumes Service, Azure NetApp Files, FSxN for NetApp ONTAP storage systems, all of which can provide persistent storage for containerized applications.

For more information visit the NetApp website [here](#).

**NetApp storage integrations**

Astra Trident is an open-source and fully-supported storage orchestrator for containers and Kubernetes distributions, including Anthos.

For more information, visit the Astra Trident website [here](#).

**Advanced configuration options**

This section is dedicated to customizations that real world users would likely need to perform when deploying this solution into production, such as creating a dedicated private image registry or deploying custom load balancer instances.

**Current support matrix for validated releases**

See [here](#) for the support matrix for validated releases.

**Anthos Overview**

Anthos with NetApp is a verified, best-practice hybrid cloud architecture for the deployment of an on-premises Google Kubernetes Engine (GKE) environment in a reliable and dependable manner. This NetApp Verified Architecture reference document serves as both a design guide and a deployment validation of the Anthos with NetApp solution deployed to bare metal and virtual environments. The architecture described in this document has been validated by subject matter experts at NetApp and Google Cloud to provide the advantages of running Anthos within your enterprise data-center environment.

**Anthos**

Anthos is a hybrid-cloud Kubernetes data center solution that enables organizations to construct and manage modern hybrid-cloud infrastructures while adopting agile workflows focused on application development.
Anthos on VMware, a solution built on open-source technologies, runs on-premises in a VMware vSphere-based infrastructure, which can connect and interoperate with Anthos GKE in Google Cloud. Adopting containers, service mesh, and other transformational technologies enables organizations to experience consistent application development cycles and production-ready workloads in local and cloud-based environments. The following figure depicts the Anthos solution and how a deployment in an on-premises data center interconnects with infrastructure in the cloud.

For more information about Anthos, see the Anthos website located here.

Anthos provides the following features:

• **Anthos configuration management.** Automates the policy and security of hybrid Kubernetes deployments.

• **Anthos Service Mesh.** Enhances application observability, security, and control with an Istio-powered service mesh.

• **Google Cloud Marketplace for Kubernetes Applications.** A catalog of curated container applications available for easy deployment.

• **Migrate for Anthos.** Automatic migration of physical services and VMs from on-premises to the cloud.

• **Stackdriver.** Management service offered by Google for logging and monitoring cloud instances.

### Deployment methods for Anthos

**Anthos clusters on VMware**

Anthos clusters deployed to VMware vSphere environments are easy to deploy, maintain, and scale rapidly for most end-user Kubernetes workloads.

For more information about Anthos clusters on VMware, deployed with NetApp, please visit the page here.
**Anthos on bare metal**

Anthos clusters deployed on bare metal servers are hardware agnostic and allow you to select a compute platform optimized for your personalized use case.

For more information about Anthos on bare metal clusters deployed with NetApp, visit here.

**Anthos Clusters on VMware**

Anthos clusters on VMware is an extension of Google Kubernetes Engine that is deployed in an end user’s private data center. An organization can deploy the same applications designed to run in containers in Google Cloud in Kubernetes clusters on-premises.

Anthos clusters on VMware can be deployed into an existing VMware vSphere environment in your data center, which can save on capital expenses and enable more rapid deployment and scaling operations.

The deployment of Anthos clusters on VMware includes the following components:

- **Anthos admin workstation.** A deployment host from which gkectl and kubectl commands can be run to deploy and interact with Anthos deployments.

- **Admin cluster.** The initial cluster deployed when setting up Anthos clusters on VMware. This cluster manages all subordinate user cluster actions, including deployment, scaling, and upgrade.

- **User cluster.** Each user cluster is deployed with it’s own load balancer instance or partition, allowing it to act as a standalone Kubernetes cluster for individual users or groups, helping to achieve full multitenancy.

The following graphic is a description of an Anthos-clusters-on-VMware deployment.
Benefits

Anthos clusters on VMware offers the following benefits:

- **Advanced multitenancy.** Each end user can be assigned their own user cluster, deployed with the virtual resources necessary for their own development environment.

- **Cost savings.** End users can realize significant cost savings by deploying multiple user clusters to the same physical environment and utilizing their own physical resources for their application deployments instead of provisioning resources in their Google Cloud environment or on large bare-metal clusters.

- **Develop then publish.** On-premises deployments can be used while applications are in development, which allows for testing of applications in the privacy of a local data center before being made publicly available in the cloud.

- **Security requirements.** Customers with increased security concerns or sensitive data sets that cannot be stored in the public cloud are able to run their applications from the security of their own data centers, thereby meeting organizational requirements.

VMware vSphere

VMware vSphere is a virtualization platform for centrally managing a large number of virtualized servers and networks running on the ESXi hypervisor.

For more information about VMware vSphere, see the [VMware vSphere website](https://www.vmware.com/products/vsphere.html).

VMware vSphere provides the following features:

- **VMware vCenter Server.** VMware vCenter Server provides unified management of all hosts and VMs from a single console and aggregates performance monitoring of clusters, hosts, and VMs.

- **VMware vSphere vMotion.** VMware vCenter allows you to hot migrate VMs between nodes in the cluster upon request in a non-disruptive manner.

- **vSphere High Availability.** To avoid disruption in the event of host failures, VMware vSphere allows hosts to be clustered and configured for high availability. VMs that are disrupted by host failure are rebooted shortly on other hosts in the cluster, restoring services.

- **Distributed Resource Scheduler (DRS).** A VMware vSphere cluster can be configured to load balance the resource needs of the VMs it is hosting. VMs with resource contentions can be hot migrated to other nodes in the cluster to make sure that enough resources are available.

Hardware requirements

Compute

Google Cloud periodically requests updated validation of partner server platforms with new releases of Anthos through their Anthos Ready platform partner program. A listing of currently validated server platforms and the versions of Anthos supported can be found [here](https://cloud.google.com/anthos/)

Operating system

Anthos clusters on VMware can be deployed to both vSphere 7 and 8 environments as chosen by the customer to help match their current datacenter infrastructure.

The following table contains a list vSphere versions that have been used by NetApp and our partners to validate the solution.
Additional hardware

To complete the deployment of Anthos with NetApp as a fully validated solution, additional data center components for networking and storage have been tested by NetApp and our partner engineers.

The following table includes information about these additional infrastructure components.

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Hardware Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mellanox</td>
<td>switch (data network)</td>
</tr>
<tr>
<td>Cisco</td>
<td>switch (management network)</td>
</tr>
<tr>
<td>NetApp</td>
<td>AFF Storage System</td>
</tr>
</tbody>
</table>

Additional software

The following table includes a list of software versions deployed in the validation environment.

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Software Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetApp</td>
<td>ONTAP</td>
<td>9.12.1</td>
</tr>
<tr>
<td>NetApp</td>
<td>Astra Trident</td>
<td>24.02.0</td>
</tr>
</tbody>
</table>

During the Anthos Ready platform validation performed by NetApp, the lab environment was built based on the following diagram, which allowed us to test multiple scenarios using various NetApp ONTAP storage backends.
Network infrastructure support resources

The following infrastructure should be in place prior to the deployment of Anthos:

- At least one DNS server providing full host-name resolution that is accessible from the in-band management network and the VM network.
- At least one NTP server that is accessible from the in-band management network and the VM network.
- A DHCP server available to provide network address leases on demand should clusters need to scale dynamically.
- (Optional) Outbound internet connectivity for both the in-band management network and the VM network.

Best practices for production deployments

This section lists several best practices that an organization should take into consideration before deploying this solution into production.

Deploy Anthos to an ESXi cluster of at least three nodes

Although it is possible to install Anthos in a vSphere cluster of less than three nodes for demonstration or evaluation purposes, this is not recommended for production workloads. Although two nodes allow for basic HA and fault tolerance, an Anthos cluster configuration must be modified to disable default host affinity, and this deployment method is not supported by Google Cloud.
Configure virtual machine and host affinity

Distributing Anthos cluster nodes across multiple hypervisor nodes can be achieved by enabling VM and host affinity.

Affinity or anti-affinity is a way to define rules for a set of VMs and/or hosts that determine whether the VMs run together on the same host or hosts in the group or on different hosts. It is applied to VMs by creating affinity groups that consist of VMs and/or hosts with a set of identical parameters and conditions. Depending on whether the VMs in an affinity group run on the same host or hosts in the group or separately on different hosts, the parameters of the affinity group can define either positive affinity or negative affinity.

To configure affinity groups, see the appropriate link below for your version of VMWare vSphere.

vSphere 6.7 Documentation: Using DRS Affinity Rules.
vSphere 7.0 Documentation: Using DRS Affinity Rules.

Anthos has a config option in each individual cluster.yaml file to automatically create node affinity rules that can be enabled or disabled based on the number of ESXi hosts in your environment.

Anthos on bare metal

Benefits

The hardware-agnostic capabilities of Anthos on bare metal allow you to select a compute platform optimized for your personalized use case and also provide many additional benefits.

Examples include the following:

• **Bring your own server.** You can use servers that match your existing infrastructure to reduce capital expenditure and management costs.

• **Bring your own Linux OS.** By choosing the Linux OS that you wish to deploy your Anthos-on-bare-metal environment to, you can ensure that the Anthos environment fits neatly into your existing infrastructure and management schemes.

• **Improved performance and lowered cost.** Without the requirement of a hypervisor, Anthos-on-bare-metal clusters call for direct access to server hardware resources, including performance-optimized hardware devices like GPUs.

• **Improved network performance and lowered latency.** Because the Anthos-on-bare-metal server nodes are directly connected to your network without a virtualized abstraction layer, they can be optimized for low latency and performance.

Hardware requirements

Compute

Google Cloud periodically requests updated validation of partner server platforms with new releases of Anthos through their Anthos Ready platform partner program. A listing of currently validated server platforms and the versions of Anthos supported can be found here.

The following table contains server platforms that have been tested by NetApp and NetApp partner engineers for the validation of Anthos on bare metal deployments.
Operating System

Anthos-on-bare-metal nodes can be configured with several different Linux distributions as chosen by the customer to help match their current datacenter infrastructure.

The following table contains a list of Linux operating systems that have been used by NetApp and our partners to validate the solution.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Release</th>
<th>Anthos Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CentOS</td>
<td>8.4.2105</td>
<td>1.14</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux</td>
<td>8.4</td>
<td>1.14</td>
</tr>
<tr>
<td>Ubuntu</td>
<td>18.04.5 LTS (with kernel 5.4.0-81-generic)</td>
<td>1.14</td>
</tr>
<tr>
<td>Ubuntu</td>
<td>20.04.2 LTS</td>
<td>1.14</td>
</tr>
</tbody>
</table>

Additional hardware

To complete the deployment of Anthos on bare metal as a fully validated solution, additional data center components for networking and storage have been tested by NetApp and our partner engineers.

The following table includes information about these additional infrastructure components.

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Hardware Name</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco</td>
<td>Nexus</td>
<td>C9336C-FX2</td>
</tr>
<tr>
<td>NetApp</td>
<td>AFF</td>
<td>A250, A220</td>
</tr>
</tbody>
</table>

Additional software

The following table includes a list of additional software versions deployed in the validation environment.

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Software name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco</td>
<td>NXOS</td>
<td>9.3(5)</td>
</tr>
<tr>
<td>NetApp</td>
<td>ONTAP</td>
<td>9.11.1P4</td>
</tr>
<tr>
<td>NetApp</td>
<td>Astra Trident</td>
<td>23.01.0</td>
</tr>
</tbody>
</table>

During the Anthos Ready platform validation performed by NetApp and our partner team at World Wide Technology (WWT), the lab environment was built based on the following diagram, which allowed us to test the functionality of each server type, operating system, the network devices, and storage systems deployed in the solution.
This multi-OS environment shows interoperability with supported OS versions for the Anthos-on-bare-metal solution. We anticipate that customers will standardize on one or a subset of operating systems for their deployment.

**Infrastructure support resources**

The following infrastructure should be in place prior to the deployment of Anthos on bare metal:

- At least one DNS server that provides a full host-name resolution accessible from the management network.
- At least one NTP server that is accessible from the management network.
- (Optional) Outbound internet connectivity for both the in-band management network.

There is a demo video of an Anthos on bare metal deployment in the Videos and Demos section of this document.

**NetApp Storage Overview**

NetApp has several storage platforms that are qualified with our Astra Trident Storage
Orchestrator to provision storage for applications deployed as containers.

- AFF and FAS systems run NetApp ONTAP and provide storage for both file-based (NFS) and block-based (iSCSI) use cases.
- Cloud Volumes ONTAP and ONTAP Select provide the same benefits in the cloud and virtual space respectively.
- NetApp Cloud Volumes Service (GCP) and Azure NetApp Files provide file-based storage in the cloud.
- Amazon FSx for NetApp ONTAP is a fully managed service on AWS that provides storage for file-based use cases.

Each storage system in the NetApp portfolio can ease both data management and movement between on-premises sites and the cloud, ensuring that your data is where your applications are.

**NetApp ONTAP**

NetApp ONTAP is a powerful storage-software tool with capabilities such as an intuitive GUI, REST APIs with automation integration, AI-informed predictive analytics and corrective action, nondisruptive hardware upgrades, and cross-storage import.

For more information about the NetApp ONTAP storage system, visit the [NetApp ONTAP website](https://www.netapp.com/products/ontap).
compression, deduplication, and compaction.

- Workload-based, QoS-controlled storage.

- Seamless integration with a public cloud for tiering and protection of data. ONTAP also provides robust data protection capabilities that sets it apart in any environment:
  - **NetApp Snapshot copies.** A fast, point-in-time backup of data using a minimal amount of disk space with no additional performance overhead.
  - **NetApp SnapMirror.** Mirrors the Snapshot copies of data from one storage system to another. ONTAP supports mirroring data to other physical platforms and cloud-native services as well.
  - **NetApp SnapLock.** Efficiently administration of nonrewritable data by writing it to special volumes that cannot be overwritten or erased for a designated period.
  - **NetApp SnapVault.** Backs up data from multiple storage systems to a central Snapshot copy that serves as a backup to all designated systems.
  - **NetApp SyncMirror.** Provides real-time, RAID-level mirroring of data to two different plexes of disks that are connected physically to the same controller.
  - **NetApp SnapRestore.** Provides fast restoration of backed-up data on demand from Snapshot copies.
  - **NetApp FlexClone.** Provides instantaneous provisioning of a fully readable and writeable copy of a NetApp volume based on a Snapshot copy.

For more information about ONTAP, see the [ONTAP 9 Documentation Center](#).

*NetApp ONTAP is available on-premises, virtualized, or in the cloud.*
**NetApp AFF/FAS**

NetApp provides robust all-flash (AFF) and scale-out hybrid (FAS) storage platforms that are tailor-made with low-latency performance, integrated data protection, and multiprotocol support.

Both systems are powered by NetApp ONTAP data management software, the industry’s most advanced data-management software for highly-available, cloud-integrated, simplified storage management to deliver the enterprise-class speed, efficiency, and security your data fabric needs.

For more information about NETAPP AFF and FAS platforms, click [here](#).

**ONTAP Select**

ONTAP Select is a software-defined deployment of NetApp ONTAP that can be deployed onto a hypervisor in your environment. It can be installed on VMware vSphere or on KVM and provides the full functionality and experience of a hardware-based ONTAP system.

For more information about ONTAP Select, click [here](#).

**Cloud Volumes ONTAP**

NetApp Cloud Volumes ONTAP is a cloud-deployed version of NetApp ONTAP available to be deployed in a number of public clouds, including: Amazon AWS, Microsoft Azure, and Google Cloud.

For more information about Cloud Volumes ONTAP, click [here](#).

**NetApp Storage Integration Overview**

**Anthos Ready storage partner program.**

Google Cloud periodically requests updated validation of partner storage integrations with new releases of Anthos through their Anthos Ready storage partner program. A list of currently validated storage solutions, CSI drivers, available features, and the versions of Anthos supported can be found [here](#).

NetApp has maintained regular compliance on a quarterly basis with requests to validate our Astra Trident CSI-compliant storage orchestrator and our ONTAP storage system with versions of Anthos.

The following table contains the Anthos versions tested by NetApp and NetApp partner engineers for validation of NetApp Astra Trident CSI drivers and feature sets as a part of the Anthos Ready storage partner program:

<table>
<thead>
<tr>
<th>Deployment Type</th>
<th>Version</th>
<th>Storage System</th>
<th>Astra Trident Version</th>
<th>Protocol</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMware</td>
<td>1.28</td>
<td>ONTAP 9.12.1</td>
<td>24.02</td>
<td>NAS</td>
<td>Multiwriter, Volume Expansion, SnapShots, PVCDDataSource</td>
</tr>
<tr>
<td>VMware</td>
<td>1.28</td>
<td>ONTAP 9.12.1</td>
<td>24.02</td>
<td>SAN</td>
<td>Raw Block, Volume Expansion, SnapShots, PVCDDataSource</td>
</tr>
<tr>
<td>Vendor</td>
<td>Version</td>
<td>ONTAP Version</td>
<td>Release</td>
<td>Storage Type</td>
<td>Features</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td>----------------</td>
<td>---------</td>
<td>--------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>VMware</td>
<td>1.15</td>
<td>ONTAP 9.12.1</td>
<td>23.04</td>
<td>NAS</td>
<td>Multiwriter, Volume Expansion, SnapShots, PVCDataSource</td>
</tr>
<tr>
<td>VMware</td>
<td>1.15</td>
<td>ONTAP 9.12.1</td>
<td>23.04</td>
<td>SAN</td>
<td>Raw Block, Volume Expansion, SnapShots, PVCDataSource</td>
</tr>
<tr>
<td>VMware</td>
<td>1.14</td>
<td>ONTAP 9.12.1</td>
<td>23.01</td>
<td>NAS</td>
<td>Multiwriter, Volume Expansion, SnapShots, PVCDataSource</td>
</tr>
<tr>
<td>VMware</td>
<td>1.14</td>
<td>ONTAP 9.12.1</td>
<td>23.01</td>
<td>SAN</td>
<td>Raw Block, Volume Expansion, SnapShots, PVCDataSource</td>
</tr>
<tr>
<td>VMware</td>
<td>1.13</td>
<td>ONTAP 9.12.1</td>
<td>22.10</td>
<td>NAS</td>
<td>Multiwriter, Volume Expansion, SnapShots, PVCDataSource</td>
</tr>
<tr>
<td>VMware</td>
<td>1.13</td>
<td>ONTAP 9.12.1</td>
<td>22.10</td>
<td>SAN</td>
<td>Raw Block, Volume Expansion, SnapShots, PVCDataSource</td>
</tr>
<tr>
<td>VMware</td>
<td>1.11</td>
<td>ONTAP 9.9.1</td>
<td>22.04</td>
<td>NAS</td>
<td>Multiwriter, Volume Expansion, SnapShots</td>
</tr>
<tr>
<td>VMware</td>
<td>1.11</td>
<td>ONTAP 9.9.1</td>
<td>22.04</td>
<td>SAN</td>
<td>Raw Block, Volume Expansion, SnapShots</td>
</tr>
<tr>
<td>VMware</td>
<td>1.11</td>
<td>Element 12.3</td>
<td>22.04</td>
<td>SAN</td>
<td>Raw Block, Volume Expansion, SnapShots</td>
</tr>
<tr>
<td>bare metal</td>
<td>1.10</td>
<td>ONTAP 9.8</td>
<td>22.01</td>
<td>NAS</td>
<td>Multiwriter, Volume Expansion, SnapShots</td>
</tr>
</tbody>
</table>
NetApp storage integrations

NetApp provides a number of products to help you with orchestrating and managing persistent data in container-based environments such as Anthos.

NetApp Astra Trident is an open-source, fully-supported storage orchestrator for containers and Kubernetes distributions, including Anthos. For more information, visit the Astra Trident website [here](#).

The following pages have additional information about the NetApp products that have been validated for application and persistent-storage management in the Anthos with NetApp solution.

Astra Trident Overview

Astra Trident is a fully supported, open-source storage orchestrator for containers and Kubernetes distributions, including Anthos. Trident works with the entire NetApp storage portfolio, including NetApp ONTAP, and it also supports NFS and iSCSI connections. Trident accelerates the DevOps workflow by allowing end users to provision and manage storage from their NetApp storage systems without requiring intervention from a storage administrator.

An administrator can configure a number of storage backends based on project needs and storage system models that enable advanced storage features, including compression, specific disk types, and QoS levels that guarantee a certain level of performance. After they are defined, these backends can be used by developers in their projects to create persistent volume claims (PVCs) and to attach persistent storage to their containers on demand.
Astra Trident has a rapid development cycle and, like Kubernetes, is released four times a year.

The documentation for the latest version of Astra Trident can be found here. A support matrix for what version of Trident has been tested with which Kubernetes distribution can be found here.

Starting with the 20.04 release, Trident setup is performed by the Trident operator. The operator makes large scale deployments easier and provides additional support including self healing for pods that are deployed as a part of the Trident install.

With the 22.04 release, a Helm chart was made available to ease the installation of the Trident Operator.

For Astra Trident installation details, please see here.

Create a storage-system backend

After completing the Astra Trident Operator install, you must configure the backend for the specific NetApp storage platform you are using. Follow the link below in order to continue the setup and configuration of Astra Trident.
Create a backend.

Create a storage class

After creating the backend, you must create a storage class that Kubernetes users will specify when they want a volume. Kubernetes users provision volumes by using persistent volume claims (PVCs) that specify a storage class by name.
Follow the link below to create a storage class.
Create a storage class

Dynamically provision a volume

You must create a Kubernetes persistent volume claim (PVC) object using the storage class to dynamically provision a volume. Follow the link below to create a PVC object.
Create a PVC

Use the volume

The volume provisioned in the above step can be used by an application by mounting the volume in the pod. The link below shows an example.

Mount the volume in a pod

Advanced configuration options

Typically, the easiest-to-deploy solution is best, but, in some cases, advanced customizations are required to meet the requirements or specifications of a specific application or the environment that solution is being deployed to. To this end, the Red Hat OpenShift with NetApp solution allows for the following customizations to meet these needs.

In this section we have documented some advanced configuration options such as using third-party load balancers or creating a private registry for hosting customized container images, both of which are prerequisites for installing the NetApp Astra Control Center.

The following pages have additional information about the advanced configuration options validated in the Red Hat OpenShift with NetApp solution:

Exploring load balancer options

An application deployed in Anthos is exposed to the world by a service that is delivered by a load balancer deployed in the Anthos on-prem environment.

The following pages have additional information about load balancer options validated in the Anthos with NetApp solution:

- Installing F5 BIG-IP load balancers
- Installing MetaLB load balancers
- Installing SeeSaw load balancers

Installing F5 BIG-IP load balancers

F5 BIG-IP is an Application Delivery Controller (ADC) that offers a broad set of advanced, production-grade traffic management and security services like L4-L7 load balancing, SSL/TLS offload, DNS, firewall, and more. These services dramatically increase the availability, security, and performance of your applications.

F5 BIG-IP can be deployed and consumed in various ways, including on dedicated hardware, in the cloud, or as a virtual appliance on-premises. Refer to the documentation here to explore and deploy F5 BIG-IP.

F5 BIG-IP was the first of the bundled load balancer solutions available with Anthos On-Prem and was used in a number of the early Anthos Ready partner validations for the Anthos with NetApp solution.
F5 BIG-IP can be deployed in standalone mode or in cluster mode. For the purpose of this validation, F5 BIG-IP was deployed in standalone mode. However, for production purposes, NetApp recommends creating a cluster of BIG-IP instances to avoid a single point of failure.

An F5 BIG-IP system can be deployed on dedicated hardware, in the cloud, or as a virtual appliance on-premises with versions greater than 12.x for it to be integrated with F5 CIS. For the purpose of this document, the F5 BIG-IP system was validated as a virtual appliance, for example using the BIG-IP VE edition.

Validated releases

This solution makes use of the virtual appliance deployed in VMware vSphere. Networking for the F5 Big-IP virtual appliance can be configured in a two-armed or three-armed configuration based on your network environment. The deployment in this document is based on the two-armed configuration. Additional details on configuring the virtual appliance for use with Anthos can be found here.

The Solutions Engineering Team at NetApp have validated the releases in the following table in our lab to work with deployments of Anthos On-Prem:

<table>
<thead>
<tr>
<th>Make</th>
<th>Type</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>F5</td>
<td>BIG-IP VE</td>
<td>15.0.1-0.0.11</td>
</tr>
<tr>
<td>F5</td>
<td>BIG-IP VE</td>
<td>16.1.0-0.0.19</td>
</tr>
</tbody>
</table>

Installation

To install F5 BIG-IP, complete the following steps:

1. Download the virtual application Open Virtual Appliance (OVA) file from F5 here.

   To download the appliance, a user must register with F5. They provide a 30-day demo license for the Big-IP Virtual Edition Load Balancer. NetApp recommends a permanent 10Gbps license for the production deployment of an appliance.

2. Right-click the Infrastructure Resource Pool and select Deploy OVF Template. A wizard launches that allows you to select the OVA file that you just downloaded in Step 1. Click Next.
3. Click Next to continue through each step and accept the default values for each screen presented until you reach the storage selection screen. Select the VM_Datastore that you would like to deploy the virtual machine to, and then click Next.

4. The next screen presented by the wizard allows you to customize the virtual networks for use in the environment. Select VM_Network for the External field and select Management_Network for the Management field. Internal and HA are used for advanced configurations for the F5 Big-IP appliance and are not configured. These parameters can be left alone, or they can be configured to connect to non-infrastructure, distributed port groups. Click Next.

5. Review the summary screen for the appliance, and, if all the information is correct, click Finish to start the deployment.

6. After the virtual appliance is deployed, right-click it and power it up. It should receive a DHCP address on the management network. The appliance is Linux-based, and it has VMware Tools deployed, so you can view the DHCP address it receives in the vSphere client.

7. Open a web browser and connect to the appliance at the IP address from the previous step. The default login is admin/admin, and, after the first login, the appliance immediately prompts you to change the admin password. It then returns you to a screen where you must log in with the new credentials.
8. The first screen prompts the user to complete the Setup Utility. Begin the utility by clicking Next.

9. The next screen prompts for activation of the license for the appliance. Click Activate to begin. When prompted on the next page, paste either the 30-day evaluation license key you received when you registered for the download or the permanent license you acquired when you purchased the appliance. Click Next.

For the device to perform activation, the network defined on the management interface must be able to reach the internet.

10. On the next screen, the End User License Agreement (EULA) is presented. If the terms in the license are acceptable, click Accept.

11. The next screen counts the elapsed time as it verifies the configuration changes that have been made so far. Click Continue to resume with the initial configuration.

12. The Configuration Change window closes, and the Setup Utility displays the Resource Provisioning menu. This window lists the features that are currently licensed and the current resource allocations for the virtual appliance and each running service.
13. Clicking the Platform menu option on the left enables additional modification of the platform. Modifications include setting the management IP address configured with DHCP, setting the host name and the time zone the appliance is installed in, and securing the appliance from SSH accessibility.

14. Next click the Network menu, which enables you to configure standard networking features. Click Next to begin the Standard Network Configuration wizard.

15. The first page of the wizard configures redundancy; leave the defaults and click Next. The next page enables you to configure an internal interface on the load balancer. Interface 1.1 maps to the VMNIC labeled Internal in the OVF deployment wizard.

The spaces in this page for Self IP Address, Netmask, and Floating IP address can be filled with a non-routable IP for use as a placeholder. They can also be filled with an internal network that has been configured as a distributed port group for virtual guests if you are deploying the three-armed configuration. They must be completed to continue with the wizard.

16. The next page enables you to configure an external network that is used to map services to the pods deployed in Kubernetes. Select a static IP from the VM_Network range, the appropriate subnet mask, and a floating IP from that same range. Interface 1.2 maps to the VMNIC labeled External in the OVF deployment wizard.

17. On the next page, you can configure an internal-HA network if you are deploying multiple virtual appliances in the environment. To proceed, you must fill the Self-IP Address and the Netmask fields, and you must select interface 1.3 as the VLAN Interface, which maps to the HA network defined by the OVF template wizard.

18. The next page enables you to configure the NTP servers. Then click Next to continue to the DNS setup. The DNS servers and domain search list should already be populated by the DHCP server. Click Next to accept the defaults and continue.

19. For the remainder of the wizard, click Next to continue through the advanced peering setup, the configuration of which is beyond the scope of this document. Then click Finish to exit the wizard.

20. Create individual partitions for the Anthos admin cluster and each user cluster deployed in the environment. Click System in the menu on the left, navigate to Users, and click Partition List.

21. The displayed screen only shows the current common partition. Click Create on the right to create the first additional partition, and name it GKE-Admin. Then click Repeat, and name the partition User-Cluster-1. Click the Repeat button again to name the next partition User-Cluster-2. Finally click Finished to complete the wizard. The Partition list screen returns with all the partitions now listed.
Integration with Anthos

There is a section in each configuration file, respectively for the admin cluster, and each user cluster that you choose to deploy to configure the load balancer so that it is managed by Anthos On Prem.

The following script is a sample from the configuration of the partition for the GKE-Admin cluster. The values that need to be uncommented and modified are placed in bold text below:

```yaml
# (Required) Load balancer configuration
loadBalancer:
  # (Required) The VIPs to use for load balancing
  vips:
    # Used to connect to the Kubernetes API
    controlPlaneVIP: "10.61.181.230"
    # # (Optional) Used for admin cluster addons (needed for multi cluster features). Must
    # # be the same across clusters
    # # addonsVIP: ""
    # (Required) Which load balancer to use "F5BigIP" "Seesaw" or "ManualLB" Uncomment
    # the corresponding field below to provide the detailed spec
    kind: F5BigIP
    # # (Required when using "ManualLB" kind) Specify pre-defined nodeports
    # manualLB:
    #   # NodePort for ingress service's http (only needed for user cluster)
    #   ingressHTTPNodePort: 0
    #   # NodePort for ingress service's https (only needed for user cluster)
    #   ingressHTTPSNodePort: 0
    #   # NodePort for control plane service
    #   controlPlaneNodePort: 30968
    #   # NodePort for addon service (only needed for admin cluster)
    #   addonsNodePort: 31405
    # # (Required when using "F5BigIP" kind) Specify the already-existing partition and
    # # credentials
    f5BigIP:
      address: "172.21.224.21"
      credentials:
        username: "admin"
        password: "admin-password"
        partition: "GKE-Admin"
      # # # (Optional) Specify a pool name if using SNAT
      # # snatPoolName: ""
      # (Required when using "Seesaw" kind) Specify the Seesaw configs
      seesaw:
        # (Required) The absolute or relative path to the yaml file to use for
```
Installing MetalLB load balancers

This page lists the installation and configuration instructions for the MetalLB managed load balancer.

Installing The MetalLB Load Balancer

The MetalLB load balancer is fully integrated with Anthos Clusters on VMware and has automated deployment performed as part of the Admin and User cluster setups starting with the 1.11 release. There are blocks of text in the respective cluster.yaml configuration files that you must modify to provide load balancer info. It is self-hosted on your Anthos cluster instead of requiring the deployment of external resources like the other supported load balancer solutions. It also allows you to create an ip-pool that automatically assigns addresses with the creation of Kubernetes services of type load balancer in clusters that do not run on a cloud provider.

Integration with Anthos

When enabling the MetalLB load balancer for Anthos admin, you must modify a few lines in the loadBalancer: section that exists in the admin-cluster.yaml file. The only values that you must modify are to set the controlPlaneVIP: address and then set the kind: as MetalLB. See the following code snippet for an example:

```
# cluster
# (Required) The Virtual Router IDentifier of VRRP for the Seesaw group. Must
# be between 1-255 and unique in a VLAN.
# vrid: 0
# (Required) The IP announced by the master of Seesaw group
# masterIP: ""
# (Required) The number CPUs per machine
# cpus: 4
# (Required) Memory size in MB per machine
# memoryMB: 8192
# (Optional) Network that the LB interface of Seesaw runs in (default: cluster)
# network)
# vCenter:
  # vSphere network name
  # networkName: VM_Network
# (Optional) Run two LB VMs to achieve high availability (default: false)
# enableHA: false
```
When enabling the MetalLB load balancer for Anthos user clusters, there are two areas in each user-cluster.yaml file that you must update. First, in a manner similar to the admin-cluster.yaml file, you must modify the controlPlaneVIP:, ingressVIP:, and kind: values in the loadBalancer: section. See the following code snippet for an example:

```
loadBalancer:
  # (Required) The VIPs to use for load balancing
  vips:
    # Used to connect to the Kubernetes API
    controlPlaneVIP: "10.61.181.240"
    # Shared by all services for ingress traffic
    ingressVIP: "10.61.181.244"
    # (Required) Which load balancer to use "F5BigIP" "Seesaw" "ManualLB" or "MetalLB".
    # Uncomment the corresponding field below to provide the detailed spec
    kind: MetalLB

# (Required) The VIPs to use for load balancing
vips:
  # Used to connect to the Kubernetes API
  controlPlaneVIP: "10.61.181.230"
  # # (Optional) Used for admin cluster addons (needed for multi cluster features). Must
  # # be the same across clusters
  # addonsVIP: ""
  # (Required) Which load balancer to use "F5BigIP" "Seesaw" "ManualLB" or "MetalLB".
  # Uncomment the corresponding field below to provide the detailed spec
  kind: MetalLB
```

The ingressVIP IP address must exist within the pool of IP addresses assigned to the MetalLB load balancer later in the configuration.

You then need to navigate to the metalLB: subsection and modify the addressPools: section by naming the pool in the - name: variable. You must also create a pool of ip-addresses that MetalLB can assign to services of type LoadBalancer by providing a range to the addresses: variable.
The address pool can be provided as a range like in the example, limiting it to a number of addresses in a particular subnet, or it can be provided as a CIDR notation if the entire subnet is made available.

1. When Kubernetes services of type LoadBalancer are created, MetalLB automatically assigns an externalIP to the services and advertises the IP address by responding to ARP requests.

### Installing SeeSaw load balancers

This page lists the installation and configuration instructions for the SeeSaw managed load balancer.

SeeSaw is the default managed network load balancer installed in an Anthos Clusters on VMware environment from versions 1.6 to 1.10.

### Installing The SeeSaw load balancer

The SeeSaw load balancer is fully integrated with Anthos Clusters on VMware and has automated deployment performed as part of the Admin and User cluster setups. There are blocks of text in the `cluster.yaml` configuration files that must be modified to provide load balancer info, and then there is an additional step prior to cluster deployment to deploy the load balancer using the built-in `gkectl` tool.

SeeSaw load balancers can be deployed in HA or non-HA mode. For the purpose of this validation, the SeeSaw load balancer was deployed in non-HA mode, which is the default setting. For production purposes, NetApp recommends deploying SeeSaw in an HA configuration for fault tolerance and reliability.

### Integration with Anthos

There is a section in each configuration file, respectively for the admin cluster, and in each user cluster that you choose to deploy to configure the load balancer so that it is managed by Anthos On-Prem.

The following text is a sample from the configuration of the partition for the GKE-Admin cluster. The values that need to be uncommented and modified are placed in bold text below:
loadBalancer:
# (Required) The VIPs to use for load balancing
vips:
# Used to connect to the Kubernetes API
  controlPlaneVIP: "10.61.181.230"
# # (Optional) Used for admin cluster addons (needed for multi cluster features). Must
# # be the same across clusters
# # addonsVIP: ""
# (Required) Which load balancer to use "F5BigIP" "Seesaw" or "ManualLB". Uncomment
# the corresponding field below to provide the detailed spec
kind: Seesaw
# # (Required when using "ManualLB" kind) Specify pre-defined nodeports
# manualLB:
# # NodePort for ingress service's http (only needed for user cluster)
# ingressHTTPNodePort: 0
# # NodePort for ingress service's https (only needed for user cluster)
# ingressHTTPSNodePort: 0
# # NodePort for control plane service
# controlPlaneNodePort: 30968
# # NodePort for addon service (only needed for admin cluster)
# addonsNodePort: 31405
# # (Required when using "F5BigIP" kind) Specify the already-existing partition and
# # credentials
# f5BigIP:
# # address:
# # credentials:
# #   username:
# #   password:
# # partition:
# #   # # (Optional) Specify a pool name if using SNAT
# #   # snatPoolName: ""
# (Required when using "Seesaw" kind) Specify the Seesaw configs
seesaw:
# (Required) The absolute or relative path to the yaml file to use for IP allocation
# for LB VMs. Must contain one or two IPs.
ipBlockFilePath: "admin-seesaw-block.yaml"
# (Required) The Virtual Router IDentifier of VRRP for the Seesaw group. Must
# be between 1-255 and unique in a VLAN.
vrid: 100
# (Required) The IP announced by the master of Seesaw group
masterIP: "10.61.181.236"

# (Required) The number CPUs per machine
cpus: 1

# (Required) Memory size in MB per machine
memoryMB: 2048

# (Optional) Network that the LB interface of Seesaw runs in (default: cluster)
cluster

# network)

vCenter:

# vSphere network name

networkName: VM_Network

# (Optional) Run two LB VMs to achieve high availability (default: false)

enableHA: false

The SeeSaw load balancer also has a separate static seesaw-block.yaml file that you must provide for each cluster deployment. This file must be located in the same directory relative to the cluster.yaml deployment file, or the full path must be specified in the section above.

A sample of the admin-seesaw-block.yaml file looks like the following script:

blocks:
- netmask: "255.255.255.0"
  gateway: "10.63.172.1"
  ips:
    - ip: "10.63.172.152"
      hostname: "admin-seesaw-vm"

This file provides the gateway and netmask for the network that the load balancer provides to the underlying cluster, as well as the management IP and hostname for the virtual machine that is deployed to run the load balancer.

Solution Validation and Use Cases

The examples provided on this page are solution validations and use cases for Anthos with NetApp.

Install an application using the Google Cloud Console

Videos and Demos

The following video demonstrates some of the capabilities described in this document:

Deploying Anthos on bare metal - Anthos with NetApp

Deployment of Trident on Anthos 1.14 cluster
Where to find additional information

To learn more about the information described in this document, review the following websites:

• NetApp Documentation
  https://docs.netapp.com/

• NetApp Astra Trident Documentation

• NetApp Astra Control Center Documentation
  https://docs.netapp.com/us-en/astra-control-center/

• Anthos Clusters on VMware Documentation
  https://cloud.google.com/anthos/clusters/docs/on-prem/latest/overview

• Anthos on bare metal Documentation
  https://cloud.google.com/anthos/clusters/docs/bare-metal/latest

• VMware vSphere Documentation
  https://docs.vmware.com/

TR-4919: DevOps with NetApp Astra

Alan Cowles and Nikhil M Kulkarni, NetApp

Use cases

The DevOps with NetApp Astra solution is architected to deliver exceptional value for customers with the following use cases:

• Easy to deploy and manage applications and development environments deployed on top of supported Kubernetes distributions.

• Discussion of real-world use cases for DevOps workflows and examples of the tools and methods that NetApp can provide to make adoption and use of these methods easier.

• Exploration of how application-consistent snapshot, backups, and clones can be used to enhance the DevOps experience.

Business value

Enterprises are increasingly adopting DevOps practices to create new products, shorten release cycles, and rapidly add new features. Because of their innate agile nature, containers and microservices play a crucial role in supporting DevOps practices. However, practicing DevOps at a production scale in an enterprise environment presents its own challenges and imposes certain requirements on the underlying infrastructure, such as the following:
• High availability at all layers in the stack so that workflows are never interrupted.
• Ease of deployment and management procedures for the end user.
• API-driven and programmable infrastructure to keep up with microservices and developer agility.
• Ability to scale infrastructure independently and in an automated fashion, based on workload demands.
• Protecting applications alongside their backing persistent data sets for DevOps workflows accelerate time to market by not having to rely on redeployments or manual copying of data.

Recognizing these capabilities and challenges, this technical report outlines the process of improving and simplifying DevOps use cases for containerized applications using the wide portfolio of NetApp products.

**Technology overview**

The DevOps with NetApp solution contains the following major components:

**DevOps practices**

DevOps practices focus on automated, repeatable, and easily manageable operations that enhance the development workflow by allowing the end user to control the environment in which they are developing their code. This solution provides several examples and use cases in which NetApp technology can be of the greatest benefit to such operations.

**Container orchestration**

There are numerous container orchestration platforms in use today. Although most of these platforms are based on Kubernetes, each has pros and cons. So it is important to understand feature sets and integrations when selecting a container orchestration platform for DevOps workflows. With the NetApp Astra suite of products, we support the following platforms for full-fledged DevOps use cases:

- **Red Hat OpenShift** 4.6.8+
- **Rancher** 2.5+
- **Kubernetes** 1.20+
- **VMware Tanzu Kubernetes Grid** 1.4+
- **VMware Tanzu Kubernetes Grid Integrated Edition** 1.12.2+

**NetApp storage systems**

Unresolved directive in containers/dwn_solution_overview.adoc - include::../../_include/containers_common_intro_sections.adoc[tag=solution_overview_netapp_storage_intro]

**NetApp storage integrations**

Unresolved directive in containers/dwn_solution_overview.adoc - include::../../_include/containers_common_intro_sections.adoc[tag=solution_overview_netapp_storage_integrations_intro]

**DevOps Overview**

Over the past several years, organizations that build software have been embracing the concepts of DevOps. DevOps practices break down organizational barriers, bringing development and operations teams closer together. DevOps practices also empower the
teams to accelerate delivery, increase availability, and make services and applications more stable, thus improving the team’s productivity. In addition, adoption of an automation framework is also a key ingredient of success — from building, testing, and operating applications at scale or managing a fully automated infrastructure platform or stack. Below we discuss some primary use cases for DevOps where NetApp solutions can be implemented to help enhance the experiences that DevOps practitioners encounter during their daily practice.

**DevOps use cases**

Although DevOps does not have a single, universally accepted definition, solutions for DevOps practitioners typically contain similar constructs or ideologies that enable easy implementation, repetition, and management at scale. The following sections describe potential use cases for DevOps workflows enabled by NetApp solutions.

**Continuous Integration, Continuous Delivery, and Continuous Deployment (CI/CD)**

Continuous Integration, Continuous Delivery, and Continuous Deployment (CI/CD) is a coding philosophy that encourages developers to implement and transform their coding practices by establishing a method by which they can consistently update, test, and deploy their code in an automated fashion. The most popular method by which CI/CD is implemented in most DevOps workflows is that of the CI/CD pipeline, and there are several third-party software applications that can help achieve this.

See the following examples of popular applications that can help with CI/CD-type workflows:

- ArgoCD
- Jenkins
- Tekton

Some of the use cases included later in this technical report have been demonstrated in Jenkins, but the primary CI/CD principles can be applied to whatever tool an organization has implemented in their own
Infrastructure as code helps provision and manage IT resources through automated commands, APIs, and software development kits (SDK). This concept greatly enhance the DevOps experience by removing physical data center or resource limitations that could prevent developers from meeting their objectives.

End users often use programming languages such as Python or automation tools such as Ansible or Puppet to create automated and repeatable infrastructure scaling actions that can be called by developers when needed.

Both NetApp ONTAP and Astra Control contain public facing APIs and ansible modules or software development toolkits that make automating operations very easy to adopt and integrate into DevOps processes.

NetApp storage systems overview
NetApp ONTAP

Unresolved directive in containers/dwn_netapp_ontap.adoc -
include::../../_include/containers_common_intro_sections.adoc[tags=netapp_ontap_page]

NetApp Storage Integration Overview

Unresolved directive in containers/dwn_overview_storint.adoc -
include::../../_include/containers_common_intro_sections.adoc[tags=storage_integration_overview]

NetApp Astra Control overview

Unresolved directive in containers/dwn_overview_astra.adoc -
include::../../_include/containers_common_intro_sections.adoc[tags=astra_cc_overview]

For a detailed installation and operations guide on Astra Control Center, follow the documentation [here](#).

Astra Control Center automation

Astra Control Center has a fully functional REST API for programmatic access. Users can use any programming language or utility to interact with Astra Control REST API endpoints. To learn more about this API, see the documentation [here](#).

If you are looking for a ready-made software development toolkit for interacting with Astra Control REST APIs, NetApp provides a toolkit with Astra Control Python SDK, which you can download [here](#).

If programming is not appropriate for your situation and you would like to use a configuration management tool, you can clone and run the Ansible playbooks that NetApp publishes [here](#).

Astra Trident Overview

Unresolved directive in containers/dwn_overview_trident.adoc -
include::../../_include/containers_common_intro_sections.adoc[tags=trident_overview]

Refer to the documentation [here](#) to install and use Astra Trident.

Use-case validation: DevOps with NetApp Astra

The following use cases have been validated for DevOps with NetApp Astra:

- Integrate Protection into CI/CD Pipelines with NetApp Astra Control
- Leverage Astra Control to facilitate Post-mortem Analysis and Restore the Application
- Accelerating Software Development with NetApp FlexClones

Integrate Protection into CI/CD Pipelines with NetApp Astra Control

Overview

One of the most common uses of DevOps workflows is continuous integration and continuous deployment (CI/CD) pipelines that build, integrate, and run automated test suites on applications as developers commit new code. DevOps engineers and site-reliability engineers (SREs) typically have pipelines dedicated to the various workflows for new feature development, regression testing, bug fixes, quality engineering, and other functions in the development process.
As teams increase their level of automation, the pace of change for in-production applications can feel overwhelming. Therefore, some teams prefer to protect in-production applications or services. In addition to protecting the code and container images, they also want to protect the application state, configuration data (such as Kubernetes objects and resources associated with the application), and an application’s persistent data.

In this use case, we take a closer look at a promotion-to-production pipeline that deploys a new version of an application: first into a staging environment and then into a production environment. This example applies equally to the major public clouds and also to an on-premises environment. Although we show the deployment of one version of the app, the pipeline can also be used with other strategies, such as blue/green or canary deployment. As part of the CI/CD pipeline, we’re going to protect the application by creating a complete application backup. An application-aware backup of the in-production application and its data, state, and configuration can be useful for numerous DevOps workflows.

The application used for validating this use-case was Magento, an e-commerce solution with a web-based front end; an Elasticsearch instance for search and analysis features; and a MariaDB database that tracks all the shopping inventory and transaction details. This containerized application was installed in a Red Hat OpenShift cluster. Every pod in the application used persistent volumes to store data. The persistent volumes were automatically created by NetApp Astra Trident, the Container Storage Interface–compliant storage orchestrator for Kubernetes that enables storage to be provisioned on NetApp storage systems. Further, to utilize the Astra Control Center’s application protection capabilities, the application in question was managed by Astra Control, which was then used to trigger application backups that stored the state of the application along with the data held in persistent volumes. We used the NetApp Astra Control Python SDK to automate the process of triggering application backups, which was then introduced into a CI/CD pipeline. This pipeline was created and executed using a popular CI/CD tool called [Jenkins] to automate the flow to build, protect, and deploy the application.

Let us run through the prerequisites and procedure to introduce protection in a CI/CD pipeline.

**Use-case validation prerequisites**

The following tools or platforms were deployed and configured as prerequisites:

1. Red Hat OpenShift Container Platform
2. NetApp Astra Trident installed on OpenShift with a backend to NetApp ONTAP system configured
3. A default storageclass configured, pointing to a NetApp ONTAP backend
4. NetApp Astra Control Center installed on an OpenShift cluster
5. OpenShift cluster added as a managed cluster to Astra Control Center
6. Jenkins installed on an OpenShift cluster and configured with an agent node with a Docker engine installed on it

Installing the application

Let’s start with the initial installation of the application in the staging and production environments. For the purpose of this use case, this step is a prerequisite, so it is performed manually. The CI/CD pipeline is used for subsequent build and deploy workflows as a result of new version releases of the application.

The production environment in this use case is a namespace called `magento-prod`, and the corresponding staging environment is a namespace called `magento-staging` configured on the Red Hat OpenShift cluster.

To install the application, complete the following steps:

1. Install the Magento application using bitnami helm chart on the production environment. We use RWX PVs for Magento and Mariadb pods.

```
[netapp-user@rhel7 na_astra_control_suite]$ helm install --version 14 magento bitnami/magento -n magento-prod --create-namespace --set image.tag=2.4.1-debian-10-r11,magentoHost=10.63.172.243,persistence.magento.accessMode=ReadWriteMany,persistence.apache.accessMode=ReadWriteMany,mariadb.master.persistence.accessModes[0]=ReadWriteMany
```

Magento bitnami helm chart requires a LoadBalancer service to expose the Magento GUI service. We used MetalLB for providing an on-prem load balancer service in this example.

2. After a few minutes, verify that all pods and services are running.

```
[netapp-user@rhel7 na_astra_control_suite]$ oc get pods -n magento-prod
NAME                                                     READY   STATUS
RESTARTS   AGE
magento-9d658fd96-qrxmt                                  1/1     Running
  0        49m
magento-elasticsearch-coordinating-only-69869cc5-768rm    1/1     Running
  0        49m
magento-elasticsearch-data-0                              1/1     Running
  0        49m
magento-elasticsearch-master-0                            1/1     Running
  0        49m
magento-mariadb-0                                         1/1     Running
  0        49m
```

383
3. Repeat the same procedure for the staging environment.

Manage the Magento application in Astra Control Center

1. Navigate to Applications and select the Discovered applications tab.

2. Click the ellipsis against the Magento application in the production environment (magento-prod), and click Manage.

3. The Magento application is now managed by the Astra Control Center. All operations supported by Astra Control can be performed on the application. Note the version of the application as well.

4. Repeat the steps for managing the Magento application in the staging environment (magento-staging).

CI/CD pipeline with integrated protection

When we work with new versions of applications, we use a CI/CD pipeline to build the container image, take backups of both the staging and production environments, deploy the new version of the application to the staging environment, wait for approval to promotion to production, and then deploy the new version of the application to the production environment. To use a CI/CD pipeline, complete the following steps:

1. Log into Jenkins, and create the required credentials: one for Magento creds, one for Mariadb admin creds, and the third for Mariadb root creds.

2. Navigate to Manage Jenkins > Manage Credentials and click the appropriate domain.

3. Click Add Credentials, and set the kind to Username with password and scope set to Global. Enter the username, password, and an ID for the credentials and click OK.
4. Repeat the same procedure for the other two credentials.
5. Go back to the Dashboard, create a pipeline by clicking New Item, and then click Pipeline.
6. Copy the pipeline from the Jenkinsfile here.
7. Paste the pipeline into the Jenkins pipeline section and then click Save.
8. Fill the parameters of the Jenkins pipeline with the respective details including the helm chart version, the Magento application version to be upgraded to, the Astra toolkit version, the Astra Control Center FQDN, the API token, and its instance ID. Specify the docker registry, namespace, and Magento IP of both production and staging environments, and also specify the credential IDs of the credentials created.

```
MAGENTO_VERSION = '2.4.1-debian-10-r14'
CHART_VERSION = '14'
RELEASE_TYPE = 'MINOR'
ASTRA_TOOLKIT_VERSION = '2.0.2'
ASTRA_API_TOKEN = 'xxxxxxxxx'
ASTRA_INSTANCE_ID = 'xxx-xxx-xxx-xxx-xxx'
ASTRA_FQDN = 'netapp-astra-control-center.org.example.com'
DOCKER_REGISTRY = 'docker.io/netapp-solutions-cicd'
PROD_NAMESPACE = 'magento-prod'
PROD_MAGENTO_IP = 'x.x.x.x'
STAGING_NAMESPACE = 'magento-staging'
STAGING_MAGENTO_IP = 'x.x.x.x'
MAGENTO_CREDS = credentials('magento-cred')
MAGENTO_MARIADB_CREDS = credentials('magento-mariadb-cred')
MAGENTO_MARIADB_ROOT_CREDS = credentials('magento-mariadb-root-cred')
```

9. Click Build Now. The pipeline starts executing and progresses through the steps. The application image is first built and uploaded to the container registry.

10. The application backups are initiated via Astra Control.
11. After the backup stages have completed successful, verify the backups from the Astra Control Center.

12. The new version of the application is then deployed to the staging environment.
13. After this step is completed, the program waits for the user to approve deployment to production. At this stage, assume that the QA team performs some manual testing and approves production. You can then click Approve to deploy the new version of the application to the production environment.

14. Verify that the production application is also upgraded to the desired version.

As part of the CI/CD pipeline, we demonstrated the ability to protect the application by creating a complete application-aware backup. Because the entire application has been backed up as part of the promotion-to-production pipeline, you can feel more confident about highly automated application deployments. This application-aware backup containing the data, state, and configuration of the application can be useful for numerous DevOps workflows. One important workflow would be to roll back to the previous version of the application in case of unforeseen issues.
Although we demonstrated a CI/CD workflow through with Jenkins tool, the concept can easily and efficiently be extrapolated to different tools and strategies. To see this use case in action, watch the video here.

**Use Astra Control to facilitate post-mortem analysis and restore the application**

**Overview**

In the first use case, we demonstrated how to use NetApp Astra Control Center to protect your applications in Kubernetes. That section describes how to integrate application backups via Astra Control directly into your development workflow by using the Python SDK in the NetApp Astra toolkit. This approach allows for the protection of development and production environments by automating on-demand backups during the continuous integration and continuous deployment (CI/CD) process. With this extra layer of application-consistent data protection added to the CI/CD pipeline and the production applications, the development processes is safe if something goes wrong in the process, which promotes good business-continuity practices.

In a traditional workflow, after encountering a failure when the application is upgraded to a new version, the development team would attempt to troubleshoot the issue in real time based on bug reports being provided by customers. Alternatively, at the first sign of trouble, the team could attempt to redeploy the application to a parallel debugging environment to take that process offline. They could redeploy an older code base from a previous version into production, which would restore the application to working order.

Although this approach works, the team would have to make sure that the state of the broken production app matched that of the version seen in production when the issues occurred. They would also have to spend time promoting the known-good build into production by fetching code from their repository and redeploying the machine images to restore the application to a good running state. Also, in this scenario, we didn’t consider whether the production database itself was corrupted by the faulty code. Ideally, there are separate backup processes in place for the database data, but must we assume that they’re consistent with the state of the application as it was published? This is where the benefits of stateful and application-consistent backups, restores and clones with Astra Control really show their value.

First, we can use Astra Control to facilitate post-mortem analysis on the state of the application. We do this by cloning the buggy production version to a parallel testing environment in an application-consistent manner. Having this environment set aside in its bug-ridden state enable us to troubleshoot the problem in real time.

Furthermore, Astra Control supports the in-place restore capability that allows us to restore the production
application to a last acceptable backup (that preceded the afflicted version of code). The restored version assumes the position of the previous, buggy production application, in an application-consistent and stateful manner, including the ingress IP previously assigned. As a result, customers accessing the front end would be unaware of the transition to the backup version.

Use-case validation prerequisites

The following tools or platforms were deployed and configured as prerequisites:

- Red Hat OpenShift Container Platform.
- NetApp Astra Trident installed on OpenShift with a backend configured to a NetApp ONTAP system.
- A default storageclass configured, pointing to a NetApp ONTAP backend.
- NetApp Astra Control Center installed on an OpenShift cluster.
- OpenShift cluster added as a managed cluster to Astra Control Center.
- Jenkins installed on an OpenShift cluster.
- Magento application installed in the production environment. The production environment in this use case is a namespace called ‘magento-prod’ in a Red Hat OpenShift cluster.
- Production application managed by Astra Control Center.
- Known-good backup(s) of the production application captured with Astra Control.

Clone and restore pipeline

Considering that the application has been upgraded to a new version, the application in the production environment (magento-prod) isn’t behaving as intended after the upgrade. Let’s assume that the data being returned by front-end queries doesn’t match the request or that the database has in fact been corrupted. To clone and restore the pipeline, complete the following steps:
1. Log into Jenkins and create a pipeline by clicking New Item and then Pipeline.

2. Copy the pipeline from the Jenkinsfile here.

3. Paste the pipeline into the Jenkins pipeline section and then click Save.

4. Fill the parameters of the Jenkins pipeline with the respective details like the current Magento application version in production, the Astra Control Center FQDN, the API token, the instance ID and application name or namespace of production and debug environments, and the source and destination cluster names. For the purpose of this use case, the production environment is a namespace called 'magento-prod' and the debug environment is a namespace called 'magento-debug' configured on a Red Hat OpenShift cluster.

   ```
   MAGENTO_VERSION = '2.4.1-debian-10-r14'
   ASTRAS_TOOLKIT_VERSION = '2.0.2'
   ASTRAS_API_TOKEN = 'xxxxxxx'
   ASTRAS_INSTANCE_ID = 'xxx-xxx-xxx-xxx-xxx'
   ASTRAS_FQDN = 'netapp-astra-control-center.org.example.com'
   PROD_APP_NAME = 'magento-prod'
   DEBUG_APP_NAME = 'magento-debug'
   DEBUG_NAMESPACE = 'magento-debug'
   PROD_KUBERNETES_CLUSTER = 'ocp-vmw'
   DEBUG_KUBERNETES_CLUSTER = 'ocp-vmw'
   ```

5. Click Build Now. The pipeline starts executing and progresses through the steps. The application is first cloned in the current state to a debug environment, and the application is then restored to the known-working backup.
6. Verify that the cloned application is the bug-containing version.

7. Verify that the production environment is restored to a working backup, and the application in production works as expected.
These two operations in tandem expedite the return to normal business operations. To see this use case in action, watch the video here.

**Accelerating software development with NetApp FlexClone technology**

**Overview**

Cloning a deployed application in a Kubernetes cluster is a very useful tool for developers that would like to expedite their workflows by sharing environments with partners or by testing new versions of code in a development environment without interfering with the version they are currently working on. The stateful and application-consistent cloning of a Kubernetes application is a major feature included with NetApp Astra Control, alongside the backup and restore of applications. As a bonus, if an application is cloned within the same Kubernetes cluster using the same storage backend, Astra Control defaults to using NetApp FlexClone technology for the duplication of persistent data volumes, speeding up the process significantly. By accelerating this process, the cloned environment is provisioned and available for use in a few moments, allowing developers to resume their work with just a brief pause when compared to redeploying their test or development environment. As an additional convenience, all of the functions available in NetApp Astra Control can be called with an API, which allows for easy integration into automation frameworks like Ansible. Therefore, environments can be staged even more rapidly because only minor changes are needed in a playbook or role to begin the cloning procedure.

**What is NetApp FlexClone technology?**

NetApp FlexClone technology is a writeable, point-in-time snapshot-based copy of a NetApp FlexVol. They are provisioned almost instantly, contain all of the data from the source volume, and consume no additional storage space until the data in the new volume begins to diverge from the source. They are often used in development or template-based environments when multiple copies of data are useful for staging purposes and storage systems have limited resources for provisioning these volumes. Compared to a traditional storage system in which data must be copied multiple times resulting in the consumption of significant storage space and time, NetApp FlexClone technology accelerates storage-dependant tasks.
To find out more about NetApp FlexClone technology, visit the page on NetApp Docs.

**Prerequisites**

1. A supported Kubernetes Distribution, such as Red Hat OpenShift 4.6.8+, Rancher 2.5+, or Kubernetes 1.19+.
3. A NetApp ONTAP system with a storage backend configured through NetApp Astra Trident.
4. Ansible 2.9+.
5. Templates for the environments that you'd like to clone as managed applications in NetApp Astra Control.

**Use-case introduction**

For this use case, we visualize something similar to the following workflow:
1. A user runs the ansible playbook to create a new staging environment.
2. Ansible uses the URI-API module to call out to Astra Control to execute the cloning operation.
3. Astra Control executes a cloning operation on a preprovisioned template environment, thus creating a new managed application.
   
   This environment can be a single standalone application in development or an entire development environment like a Jenkins CI/CD pipeline.

4. The user then pulls a version of their code into the cloned dev environment from an online repository like Gitea.
5. The new version of the application is deployed and managed by NetApp Astra Control.
   
   Both of these processes can be automated.

6. The user can develop new code in this cloned environment.
7. When the user is satisfied with their development efforts, they can push the code back to the hosted repository.

The use case presented here depends on the existence of golden templates for the particular environments or applications you would like to clone. In our environment we have created three such templates, one for a Wordpress deployment, one for a Magento deployment, and one for a Jenkins CI/CD environment with Gitea that we have titled DevTools.
Each of these environments is managed by NetApp Astra control, with persistent volumes currently stored on a NetApp ONTAP storage system with an NFS backend provided by NetApp Astra Trident.

**Use-case validation**

1. Clone the ansible toolkit provided by the NetApp Solutions Engineering team, which includes the cloning role and the application update playbook.

   ```bash
   [netapp-user@rhel7 ~]$ git clone https://github.com/NetApp-Automation/na_astra_control_suite.git
   [netapp-user@rhel7 ~]$ cd na_astra_control_suite
   ```

2. Edit `vars/clone_vars.yml` and fill in the global values that fit your Astra Control environment.

   ```yaml
   astra_control_fqdn: astra-control-center.example.com
   astra_control_account_id: "xxxx-xxxx-xxxx-xxxx-xxxx"
   astra_control_api_token: "xxxxx"
   ```
3. With the global variables completed, you can choose the values for the specific application you wish to clone. To clone the devtools environment to a personal environment called alan-devtools, you would do the following:

```yaml
clone_details:
  - clone_name: alan-devtools
    destination_namespace: alan-dev-namespace
    source_cluster_name: ocp-vmware2
    destination_cluster_name: ocp-vmware2
    source_application_name: devtools-template
```

To take advantage of NetApp FlexClone technology in the cloning process, `src-cluster` and `dest-cluster` must be the same.

4. You can now execute the playbook to clone the application.
The playbook as written must be run by the root user or someone that can escalate through the sudo process by passing the "-K" argument.

5. When the playbook completes its run, the cloned application shows as available in the Astra Control Center console.

6. A user can then log into the Kubernetes environment where the application was deployed, verify that the application is exposed with a new IP address, and start their development work.

For a demonstration of this use case and an example of upgrading an application, see here.

**Videos and demos: DevOps with NetApp Astra**

The following videos demonstrate some of the capabilities described in this document:

- Data Protection in CI/CD pipeline with Astra Control Center
- Leverage NetApp Astra Control to Perform Post-mortem Analysis and Restore Your Application
Accelerate Software Development with Astra Control and NetApp FlexClone Technology

Additional Information: DevOps with NetApp Astra

To learn more about the information described in this document, review the following websites:

- NetApp Documentation
  https://docs.netapp.com/
- Astra Trident Documentation
  https://docs.netapp.com/us-en/trident/
- NetApp Astra Control Center Documentation
  https://docs.netapp.com/us-en/astra-control-center/
- Ansible Documentation
  https://docs.ansible.com/
- Red Hat OpenShift Documentation
  https://access.redhat.com/documentation/en-us/openshift_container_platform/4.8/
- Rancher Documentation
  https://rancher.com/docs/
- Kubernetes Documentation
  https://kubernetes.io/docs/home/

NVA-1160: Red Hat OpenShift with NetApp

Alan Cowles and Nikhil M Kulkarni, NetApp

This reference document provides deployment validation of the Red Hat OpenShift solution, deployed through Installer Provisioned Infrastructure (IPI) in several different data center environments as validated by NetApp. It also details storage integration with NetApp storage systems by making use of the Astra Trident storage orchestrator for the management of persistent storage. Lastly, a number of solution validations and real world use cases are explored and documented.

Use cases

The Red Hat OpenShift with NetApp solution is architected to deliver exceptional value for customers with the following use cases:

- Easy to deploy and manage Red Hat OpenShift deployed using IPI (Installer Provisioned Infrastructure) on bare metal, Red Hat OpenStack Platform, Red Hat Virtualization, and VMware vSphere.
- Combined power of enterprise container and virtualized workloads with Red Hat OpenShift deployed
virtually on OSP, RHV, or vSphere, or on bare metal with OpenShift Virtualization.

- Real world configuration and use cases highlighting the features of Red Hat OpenShift when used with NetApp storage and Astra Trident, the open source storage orchestrator for Kubernetes.

Business value

Enterprises are increasingly adopting DevOps practices to create new products, shorten release cycles, and rapidly add new features. Because of their innate agile nature, containers and microservices play a crucial role in supporting DevOps practices. However, practicing DevOps at a production scale in an enterprise environment presents its own challenges and imposes certain requirements on the underlying infrastructure, such as the following:

- High availability at all layers in the stack
- Ease of deployment procedures
- Non-disruptive operations and upgrades
- API-driven and programmable infrastructure to keep up with microservices agility
- Multitenancy with performance guarantees
- Ability to run virtualized and containerized workloads simultaneously
- Ability to scale infrastructure independently based on workload demands

Red Hat OpenShift with NetApp acknowledges these challenges and presents a solution that helps address each concern by implementing the fully automated deployment of RedHat OpenShift IPI in the customer’s choice of data center environment.

Technology overview

The Red Hat OpenShift with NetApp solution is comprised of the following major components:

Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform is a fully supported enterprise Kubernetes platform. Red Hat makes several enhancements to open-source Kubernetes to deliver an application platform with all the components fully integrated to build, deploy, and manage containerized applications.

For more information visit the OpenShift website [here](#).

NetApp storage systems

NetApp has several storage systems perfect for enterprise data centers and hybrid cloud deployments. The NetApp portfolio includes NetApp ONTAP, NetApp Element, and NetApp e-Series storage systems, all of which can provide persistent storage for containerized applications.

For more information visit the NetApp website [here](#).

NetApp storage integrations

NetApp Astra Control Center offers a rich set of storage and application-aware data management services for stateful Kubernetes workloads, deployed in an on-prem environment and powered by trusted NetApp data protection technology.

For more information, visit the NetApp Astra website [here](#).
Astra Trident is an open-source and fully-supported storage orchestrator for containers and Kubernetes distributions, including Red Hat OpenShift.

For more information, visit the Astra Trident website here.

Advanced configuration options

This section is dedicated to customizations that real world users would likely need to perform when deploying this solution into production, such as creating a dedicated private image registry or deploying custom load balancer instances.

Current support matrix for validated releases

<table>
<thead>
<tr>
<th>Technology</th>
<th>Purpose</th>
<th>Software version</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetApp ONTAP</td>
<td>Storage</td>
<td>9.8, 9.9.1</td>
</tr>
<tr>
<td>NetApp Element</td>
<td>Storage</td>
<td>12.3</td>
</tr>
<tr>
<td>NetApp Astra Control Center</td>
<td>Application Aware Data Management</td>
<td>21.12.60</td>
</tr>
<tr>
<td>NetApp Astra Trident</td>
<td>Storage Orchestration</td>
<td>22.01.0</td>
</tr>
<tr>
<td>Red Hat OpenShift</td>
<td>Container orchestration</td>
<td>4.6 EUS, 4.7, 4.8</td>
</tr>
<tr>
<td>Red Hat OpenStack Platform</td>
<td>Private Cloud Infrastructure</td>
<td>16.1</td>
</tr>
<tr>
<td>Red Hat Virtualization</td>
<td>Data center virtualization</td>
<td>4.4</td>
</tr>
<tr>
<td>VMware vSphere</td>
<td>Data center virtualization</td>
<td>6.7U3</td>
</tr>
</tbody>
</table>

OpenShift Overview

The Red Hat OpenShift Container Platform unites development and IT operations on a single platform to build, deploy, and manage applications consistently across on-premises and hybrid cloud infrastructures. Red Hat OpenShift is built on open-source innovation and industry standards, including Kubernetes and Red Hat Enterprise Linux CoreOS, the world’s leading enterprise Linux distribution designed for container-based workloads. OpenShift is part of the Cloud Native Computing Foundation (CNCF) Certified Kubernetes program, providing portability and interoperability of container workloads.

Red Hat OpenShift provides the following capabilities:

- **Self-service provisioning** Developers can quickly and easily create applications on demand from the tools that they use most, while operations retain full control over the entire environment.

- **Persistent storage** By providing support for persistent storage, OpenShift Container Platform allows you to run both stateful applications and cloud-native stateless applications.

- **Continuous integration and continuous development (CI/CD)** This source-code platform manages build and deployment images at scale.

- **Open-source standards** These standards incorporate the Open Container Initiative (OCI) and Kubernetes for container orchestration, in addition to other open-source technologies. You are not restricted to the technology or to the business roadmap of a specific vendor.
• **CI/CD pipelines** OpenShift provides out-of-the-box support for CI/CD pipelines so that development teams can automate every step of the application delivery process and make sure it’s executed on every change that is made to the code or configuration of the application.

• **Role-Based Access Control (RBAC)** This feature provides team and user tracking to help organize a large developer group.

• **Automated build and deploy** OpenShift gives developers the option to build their containerized applications or have the platform build the containers from the application source code or even the binaries. The platform then automates deployment of these applications across the infrastructure based on the characteristic that was defined for the applications. For example, how quantity of resources that should be allocated and where on the infrastructure they should be deployed in order for them to be compliant with third-party licenses.

• **Consistent environments** OpenShift makes sure that the environment provisioned for developers and across the lifecycle of the application is consistent from the operating system, to libraries, runtime version (for example, Java runtime), and even the application runtime in use (for example, tomcat) in order to remove the risks originated from inconsistent environments.

• **Configuration management** Configuration and sensitive data management is built in to the platform to make sure that a consistent and environment agnostic application configuration is provided to the application no matter which technologies are used to build the application or which environment it is deployed.

• **Application logs and metric.** Rapid feedback is an important aspect of application development. OpenShift integrated monitoring and log management provides immediate metrics back to developers in order for them to study how the application is behaving across changes and be able to fix issues as early as possible in the application lifecycle.

• **Security and container catalog** OpenShift offers multitenancy and protects the user from harmful code execution by using established security with Security-Enhanced Linux (SELinux), CGroups, and Secure Computing Mode (seccomp) to isolate and protect containers. It also provides encryption through TLS certificates for the various subsystems and access to Red Hat certified containers (access.redhat.com/containers) that are scanned and graded with a specific emphasis on security to provide certified, trusted, and secure application containers to end users.
Deployment methods for Red Hat OpenShift

Starting with Red Hat OpenShift 4, the deployment methods for OpenShift include manual deployments using User Provisioned Infrastructure (UPI) for highly customized deployments or fully automated deployments using Installer Provisioned Infrastructure (IPI).

The IPI installation method is the preferred method in most cases because it allows for the rapid deployment of OpenShift clusters for dev, test, and production environments.

IPI installation of Red Hat OpenShift

The Installer Provisioned Infrastructure (IPI) deployment of OpenShift involves these high-level steps:

1. Visit the Red Hat OpenShift website and login with your SSO credentials.
2. Select the environment that you would like to deploy Red Hat OpenShift into.
3. On the next screen download the installer, the unique pull secret, and the CLI tools for management.

4. Follow the installation instructions provided by Red Hat to deploy to your environment of choice.

NetApp validated OpenShift deployments

NetApp has tested and validated the deployment of Red Hat OpenShift in its labs using the Installer Provisioned Infrastructure (IPI) deployment method in each of the following data center environments:

- OpenShift on Bare Metal
- OpenShift on Red Hat OpenStack Platform
- OpenShift on Red Hat Virtualization
- OpenShift on VMware vSphere
OpenShift on Bare Metal

OpenShift on Bare Metal provides an automated deployment of the OpenShift Container Platform on commodity servers.

OpenShift on Bare Metal is similar to virtual deployments of OpenShift, which provide ease of deployment, rapid provisioning, and scaling of OpenShift clusters, while supporting virtualized workloads for applications that are not ready to be containerized. By deploying on bare metal, you do not require the extra overhead necessary to manage the host hypervisor environment in addition to the OpenShift environment. By deploying directly on bare metal servers, you can also reduce the physical overhead limitations of having to share resources between the host and OpenShift environment.

OpenShift on Bare Metal provides the following features:

• **IPI or assisted installer deployment** With an OpenShift cluster deployed by Installer Provisioned Infrastructure (IPI) on bare metal servers, customers can deploy a highly versatile, easily scalable OpenShift environment directly on commodity servers, without the need to manage a hypervisor layer.

• **Compact cluster design** To minimize the hardware requirements, OpenShift on bare metal allows for users to deploy clusters of just 3 nodes, by enabling the OpenShift control plane nodes to also act as worker nodes and host containers.

• **OpenShift virtualization** OpenShift can run virtual machines within containers by using OpenShift Virtualization. This container-native virtualization runs the KVM hypervisor inside of a container, and attaches persistent volumes for VM storage.

• **AI/ML-optimized infrastructure** Deploy applications like Kubeflow for machine learning applications by incorporating GPU-based worker nodes to your OpenShift environment and leveraging OpenShift Advanced Scheduling.

Network design

The Red Hat OpenShift on NetApp solution uses two data switches to provide primary data connectivity at 25Gbps. It also uses two management switches that provide connectivity at 1Gbps for in-band management for the storage nodes and out-of-band management for IPMI functionality.

For OpenShift bare-metal IPI deployment, you must create a provisioner node, a Red Hat Enterprise Linux 8 machine that must have network interfaces attached to separate networks.

• **Provisioning network** This network is used to boot the bare-metal nodes and install the necessary images and packages to deploy the OpenShift cluster.

• **Bare-metal network** This network is used for public-facing communication of the cluster after it is deployed.

For the setup of the provisioner node, the customer creates bridge interfaces that allow the traffic to route properly on the node itself and on the Bootstrap VM that is provisioned for deployment purposes. After the cluster is deployed, the API and ingress VIP addresses are migrated from the bootstrap node to the newly deployed cluster.

The following images depict the environment both during IPI deployment and after the deployment is complete.
VLAN requirements

The Red Hat OpenShift with NetApp solution is designed to logically separate network traffic for different purposes by using virtual local area networks (VLANs).
### VLANs

<table>
<thead>
<tr>
<th>Purpose</th>
<th>VLAN ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Out-of-band management network</td>
<td>Management for bare metal nodes and IPMI</td>
</tr>
<tr>
<td>Bare-metal network</td>
<td>Network for OpenShift services once cluster is available</td>
</tr>
<tr>
<td>Provisioning network</td>
<td>Network for PXE boot and installation of bare metal nodes via IPI</td>
</tr>
</tbody>
</table>

Although each of these networks is virtually separated by VLANs, each physical port must be set up in Access Mode with the primary VLAN assigned, because there is no way to pass a VLAN tag during a PXE boot sequence.

### Network infrastructure support resources

The following infrastructure should be in place prior to the deployment of the OpenShift container platform:

- At least one DNS server that provides a full host-name resolution accessible from the in-band management network and the VM network.
- At least one NTP server that is accessible from the in-band management network and the VM network.
- (Optional) Outbound internet connectivity for both the in-band management network and the VM network.

### OpenShift on Red Hat OpenStack Platform

The Red Hat OpenStack Platform delivers an integrated foundation to create, deploy, and scale a secure and reliable private OpenStack cloud.

OSP is an infrastructure-as-a-service (IaaS) cloud implemented by a collection of control services that manage compute, storage, and networking resources. The environment is managed using a web-based interface that allows administrators and users to control, provision, and automate OpenStack resources. Additionally, the OpenStack infrastructure is facilitated through an extensive command line interface and API enabling full automation capabilities for administrators and end-users.

The OpenStack project is a rapidly developed community project that provides updated releases every six months. Initially Red Hat OpenStack Platform kept pace with this release cycle by publishing a new release along with every upstream release and providing long term support for every third release. Recently, with the OSP 16.0 release (based on OpenStack Train), Red Hat has chosen not to keep pace with release numbers but instead has backported new features into sub-releases. The most recent release is Red Hat OpenStack Platform 16.1, which includes backported advanced features from the Ussuri and Victoria releases upstream.

For more information about OSP see the [Red Hat OpenStack Platform website](https://www.redhat.com/en/products/openstack).

### OpenStack services

OpenStack Platform services are deployed as containers, which isolates services from one another and enables easy upgrades. The OpenStack Platform uses a set of containers built and managed with Kolla. The deployment of services is performed by pulling container images from the Red Hat Custom Portal. These service containers are managed using the Podman command and are deployed, configured, and maintained with Red Hat OpenStack Director.
The Red Hat OpenShift with NetApp solution uses two data switches to provide primary data connectivity at 25Gbps. It also uses two additional management switches that provide connectivity at 1Gbps for in-band management for the storage nodes and out-of-band management for IPMI functionality.
IPMI functionality is required by Red Hat OpenStack Director to deploy Red Hat OpenStack Platform using the Ironic bare-metal provision service.

**VLAN requirements**

Red Hat OpenShift with NetApp is designed to logically separate network traffic for different purposes by using virtual local area networks (VLANs). This configuration can be scaled to meet customer demands or to provide further isolation for specific network services. The following table lists the VLANs that are required to implement the solution while validating the solution at NetApp.

<table>
<thead>
<tr>
<th>VLANs</th>
<th>Purpose</th>
<th>VLAN ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Out-of-band management network</td>
<td>Network used for management of physical nodes and IPMI service for Ironic.</td>
<td>16</td>
</tr>
<tr>
<td>Storage infrastructure</td>
<td>Network used for controller nodes to map volumes directly to support infrastructure services like Swift.</td>
<td>201</td>
</tr>
<tr>
<td>Storage Cinder</td>
<td>Network used to map and attach block volumes directly to virtual instances deployed in the environment.</td>
<td>202</td>
</tr>
<tr>
<td>Internal API</td>
<td>Network used for communication between the OpenStack services using API communication, RPC messages, and database communication.</td>
<td>301</td>
</tr>
<tr>
<td>Tenant</td>
<td>Neutron provides each tenant with their own networks via tunneling through VXLAN. Network traffic is isolated within each tenant network. Each tenant network has an IP subnet associated with it, and network namespaces mean that multiple tenant networks can use the same address range without causing conflicts.</td>
<td>302</td>
</tr>
<tr>
<td>Storage management</td>
<td>OpenStack Object Storage (Swift) uses this network to synchronize data objects between participating replica nodes. The proxy service acts as the intermediary interface between user requests and the underlying storage layer. The proxy receives incoming requests and locates the necessary replica to retrieve the requested data.</td>
<td>303</td>
</tr>
<tr>
<td>PXE</td>
<td>The OpenStack Director provides PXE boot as a part of the Ironic bare metal provisioning service to orchestrate the installation of the OSP Overcloud.</td>
<td>3484</td>
</tr>
<tr>
<td>External</td>
<td>Publicly available network which hosts the OpenStack Dashboard (Horizon) for graphical management and allows for public API calls to manage OpenStack services.</td>
<td>3485</td>
</tr>
<tr>
<td>In-band management network</td>
<td>Provides access for system administration functions such as SSH access, DNS traffic, and Network Time Protocol (NTP) traffic. This network also acts as a gateway for non-controller nodes.</td>
<td>3486</td>
</tr>
</tbody>
</table>

**Network infrastructure support resources**

The following infrastructure should be in place prior to the deployment of the OpenShift Container Platform:

- At least one DNS server which provides a full host-name resolution.
- At least three NTP servers which can keep time synchronized for the servers in the solution.
- (Optional) Outbound internet connectivity for the OpenShift environment.
Best practices for production deployments

This section lists several best practices that an organization should take into consideration before deploying this solution into production.

Deploy OpenShift to an OSP private cloud with at least three compute nodes

The verified architecture described in this document presents the minimum hardware deployment suitable for HA operations by deploying three OSP controller nodes and two OSP compute nodes. This architecture ensures a fault tolerant configuration in which both compute nodes can launch virtual instances and deployed VMs can migrate between the two hypervisors.

Because Red Hat OpenShift initially deploys with three master nodes, a two-node configuration might cause at least two masters to occupy the same node, which can lead to a possible outage for OpenShift if that specific node becomes unavailable. Therefore, it is a Red Hat best practice to deploy at least three OSP compute nodes so that the OpenShift masters can be distributed evenly and the solution receives an added degree of fault tolerance.

Configure virtual machine/host affinity

Distributing the OpenShift masters across multiple hypervisor nodes can be achieved by enabling VM/host affinity.

Affinity is a way to define rules for a set of VMs and/or hosts that determine whether the VMs run together on the same host or hosts in the group or on different hosts. It is applied to VMs by creating affinity groups that consist of VMs and/or hosts with a set of identical parameters and conditions. Depending on whether the VMs in an affinity group run on the same host or hosts in the group or separately on different hosts, the parameters of the affinity group can define either positive affinity or negative affinity. In the Red Hat OpenStack Platform, host affinity and anti-affinity rules can be created and enforced by creating server groups and configuring filters so that instances deployed by Nova in a server group deploy on different compute nodes.

A server group has a default maximum of 10 virtual instances that it can manage placement for. This can be modified by updating the default quotas for Nova.

There is a specific hard affinity/anti-affinity limit for OSP server groups; if there not enough resources to deploy on separate nodes or not enough resources to allow sharing of nodes, the VM fails to boot.

To configure affinity groups, see How do I configure Affinity and Anti-Affinity for OpenStack instances?.

Use a custom install file for OpenShift deployment

IPI makes the deployment of OpenShift clusters easy through the interactive wizard discussed earlier in this document. However, it is possible that you might need to change some default values as a part of a cluster deployment.

In these instances, you can run and task the wizard without immediately deploying a cluster; instead it creates a configuration file from which the cluster can be deployed later. This is very useful if you need to change any IPI defaults, or if you want to deploy multiple identical clusters in your environment for other uses such as multitenancy. For more information about creating a customized install configuration for OpenShift, see Red Hat OpenShift Installing a Cluster on OpenStack with Customizations.
OpenShift on Red Hat Virtualization

Red Hat Virtualization (RHV) is an enterprise virtual data center platform that runs on Red Hat Enterprise Linux (RHEL) and uses the KVM hypervisor.

For more information about RHV, see the Red Hat Virtualization website.

RHV provides the following features:

- **Centralized management of VMs and hosts** The RHV manager runs as a physical or virtual machine (VM) in the deployment and provides a web-based GUI for the management of the solution from a central interface.

- **Self-hosted engine** To minimize hardware requirements, RHV allows RHV Manager (RHV-M) to be deployed as a VM on the same hosts that run guest VMs.

- **High availability** To avoid disruption in event of host failures, RHV allows VMs to be configured for high availability. The highly available VMs are controlled at the cluster level using resiliency policies.

- **High scalability** A single RHV cluster can have up to 200 hypervisor hosts enabling it to support requirements of massive VMs to host resource-greedy, enterprise-class workloads.

- **Enhanced security** Inherited from RHV, Secure Virtualization (sVirt) and Security Enhanced Linux (SELinux) technologies are employed by RHV for the purposes of elevated security and hardening for the hosts and VMs. The key advantage from these features is logical isolation of a VM and its associated resources.

Network design

The Red Hat OpenShift on NetApp solution uses two data switches to provide primary data connectivity at 25Gbps. It also uses two additional management switches that provide connectivity at 1Gbps for in-band management of the storage nodes and out-of-band management for IPMI functionality. OCP uses the virtual machine logical network on RHV for cluster management. This section describes the arrangement and purpose of each virtual network segment used in the solution and outlines the prerequisites for deploying the solution.

**VLAN requirements**

Red Hat OpenShift on RHV is designed to logically separate network traffic for different purposes by using virtual local area networks (VLANs). This configuration can be scaled to meet customer demands or to provide...
further isolation for specific network services. The following table lists the VLANs that are required to implement the solution while validating the solution at NetApp.

<table>
<thead>
<tr>
<th>VLANs</th>
<th>Purpose</th>
<th>VLAN ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Out-of-band management network</td>
<td>Management for physical nodes and IPMI</td>
<td>16</td>
</tr>
<tr>
<td>VM Network</td>
<td>Virtual guest network access</td>
<td>1172</td>
</tr>
<tr>
<td>In-band management network</td>
<td>Management for RHV-H nodes, RHV-Manager, and ovirtmgmt network</td>
<td>3343</td>
</tr>
<tr>
<td>Storage network</td>
<td>Storage network for NetApp Element iSCSI</td>
<td>3344</td>
</tr>
<tr>
<td>Migration network</td>
<td>Network for virtual guest migration</td>
<td>3345</td>
</tr>
</tbody>
</table>

**Network infrastructure support resources**

The following infrastructure should be in place prior to the deployment of the OpenShift Container Platform:

- At least one DNS server providing full host-name resolution that is accessible from the in-band management network and the VM network.
- At least one NTP server that is accessible from the in-band management network and the VM network.
- (Optional) Outbound internet connectivity for both the in-band management network and the VM network.

**Best practices for production deployments**

This section lists several best practices that an organization should take into consideration before deploying this solution into production.

**Deploy OpenShift to an RHV cluster of at least three nodes**

The verified architecture described in this document presents the minimum hardware deployment suitable for HA operations by deploying two RHV-H hypervisor nodes and ensuring a fault tolerant configuration where both hosts can manage the hosted-engine and deployed VMs can migrate between the two hypervisors.

Because Red Hat OpenShift initially deploys with three master nodes, it is ensured in a two-node configuration that at least two masters will occupy the same node, which can lead to a possible outage for OpenShift if that specific node becomes unavailable. Therefore, it is a Red Hat best practice that at least three RHV-H hypervisor nodes be deployed as part of the solution so that the OpenShift masters can be distributed evenly and the solution receives an added degree of fault tolerance.

**Configure virtual machine/host affinity**

You can distribute the OpenShift masters across multiple hypervisor nodes by enabling VM/host affinity.

Affinity is a way to define rules for a set of VMs and/or hosts that determine whether the VMs run together on the same host or hosts in the group or on different hosts. It is applied to VMs by creating affinity groups that consist of VMs and/or hosts with a set of identical parameters and conditions. Depending on whether the VMs in an affinity group run on the same host or hosts in the group or separately on different hosts, the parameters of the affinity group can define either positive affinity or negative affinity.

The conditions defined for the parameters can be either hard enforcement or soft enforcement. Hard enforcement ensures that the VMs in an affinity group always follows the positive or negative affinity strictly without any regards to external conditions. Soft enforcement ensures that a higher preference is set for the
VMs in an affinity group to follow the positive or negative affinity whenever feasible. In the two or three hypervisor configuration described in this document, soft affinity is the recommended setting. In larger clusters, hard affinity can correctly distribute OpenShift nodes.

To configure affinity groups, see the Red Hat 6.11. Affinity Groups documentation.

**Use a custom install file for OpenShift deployment**

IPI makes the deployment of OpenShift clusters easy through the interactive wizard discussed earlier in this document. However, it is possible that there are some default values that might need to be changed as a part of cluster deployment.

In these instances, you can run and task the wizard without immediately deploying a cluster. Rather, a configuration file is created from which the cluster can be deployed later. This is very useful if you want to change any IPI defaults or if you want to deploy multiple identical clusters in your environment for other uses such as multitenancy. For more information about creating a customized install configuration for OpenShift, see Red Hat OpenShift Installing a Cluster on RHV with Customizations.

**OpenShift on VMware vSphere**

VMware vSphere is a virtualization platform for centrally managing a large number of virtualized servers and networks running on the ESXi hypervisor.

For more information about VMware vSphere, see the VMware vSphere website.

VMware vSphere provides the following features:

- **VMware vCenter Server** VMware vCenter Server provides unified management of all hosts and VMs from a single console and aggregates performance monitoring of clusters, hosts, and VMs.

- **VMware vSphere vMotion** VMware vCenter allows you to hot migrate VMs between nodes in the cluster upon request in a nondisruptive manner.

- **vSphere High Availability** To avoid disruption in the event of host failures, VMware vSphere allows hosts to be clustered and configured for High Availability. VMs that are disrupted by host failure are rebooted shortly on other hosts in the cluster, restoring services.

- **Distributed Resource Scheduler (DRS)** A VMware vSphere cluster can be configured to load balance the resource needs of the VMs it is hosting. VMs with resource contentions can be hot migrated to other nodes in the cluster to make sure that enough resources are available.
Network design

The Red Hat OpenShift on NetApp solution uses two data switches to provide primary data connectivity at 25Gbps. It also uses two additional management switches that provide connectivity at 1Gbps for in-band management for the storage nodes and out-of-band management for IPMI functionality. OCP uses the VM logical network on VMware vSphere for its cluster management. This section describes the arrangement and purpose of each virtual network segment used in the solution and outlines the prerequisites for deployment of the solution.

VLAN requirements

Red Hat OpenShift on VMware vSphere is designed to logically separate network traffic for different purposes by using virtual local area networks (VLANs). This configuration can be scaled to meet customer demands or to provide further isolation for specific network services. The following table lists the VLANs that are required to implement the solution while validating the solution at NetApp.

<table>
<thead>
<tr>
<th>VLANs</th>
<th>Purpose</th>
<th>VLAN ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Out-of-band management network</td>
<td>Management for physical nodes and IPMI</td>
<td>16</td>
</tr>
<tr>
<td>VM Network</td>
<td>Virtual guest network access</td>
<td>181</td>
</tr>
<tr>
<td>Storage network</td>
<td>Storage network for ONTAP NFS</td>
<td>184</td>
</tr>
<tr>
<td>Storage network</td>
<td>Storage network for ONTAP iSCSI</td>
<td>185</td>
</tr>
<tr>
<td>In-band management network</td>
<td>Management for ESXi Nodes, VCenter Server, ONTAP Select</td>
<td>3480</td>
</tr>
</tbody>
</table>
### VLANs

<table>
<thead>
<tr>
<th>Purpose</th>
<th>VLAN ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage network for NetApp Element iSCSI</td>
<td>3481</td>
</tr>
<tr>
<td>Network for virtual guest migration</td>
<td>3482</td>
</tr>
</tbody>
</table>

### Network infrastructure support resources

The following infrastructure should be in place prior to the deployment of the OpenShift Container Platform:

- At least one DNS server providing full host-name resolution that is accessible from the in-band management network and the VM network.
- At least one NTP server that is accessible from the in-band management network and the VM network.
- (Optional) Outbound internet connectivity for both the in-band management network and the VM network.

### Best practices for production deployments

This section lists several best practices that an organization should take into consideration before deploying this solution into production.

### Deploy OpenShift to an ESXi cluster of at least three nodes

The verified architecture described in this document presents the minimum hardware deployment suitable for HA operations by deploying two ESXi hypervisor nodes and ensuring a fault tolerant configuration by enabling VMware vSphere HA and VMware vMotion. This configuration allows deployed VMs to migrate between the two hypervisors and reboot should one host become unavailable.

Because Red Hat OpenShift initially deploys with three master nodes, at least two masters in a two-node configuration can occupy the same node under some circumstances, which can lead to a possible outage for OpenShift if that specific node becomes unavailable. Therefore, it is a Red Hat best practice that at least three ESXi hypervisor nodes must be deployed so that the OpenShift masters can be distributed evenly, which provides an added degree of fault tolerance.

### Configure virtual machine and host affinity

Ensuring the distribution of the OpenShift masters across multiple hypervisor nodes can be achieved by enabling VM and host affinity.

Affinity or anti-affinity is a way to define rules for a set of VMs and/or hosts that determine whether the VMs run together on the same host or hosts in the group or on different hosts. It is applied to VMs by creating affinity groups that consist of VMs and/or hosts with a set of identical parameters and conditions. Depending on whether the VMs in an affinity group run on the same host or hosts in the group or separately on different hosts, the parameters of the affinity group can define either positive affinity or negative affinity.

To configure affinity groups, see the [vSphere 6.7 Documentation: Using DRS Affinity Rules](#).

### Use a custom install file for OpenShift deployment

IPI makes the deployment of OpenShift clusters easy through the interactive wizard discussed earlier in this document. However, it is possible that you might need to change some default values as a part of a cluster deployment.

In these instances, you can run and task the wizard without immediately deploying a cluster, but instead the wizard creates a configuration file from which the cluster can be deployed later. This is very useful if you need
to changes any IPI defaults, or if you want to deploy multiple identical clusters in your environment for other uses such as multitenancy. For more information about creating a customized install configuration for OpenShift, see Red Hat OpenShift Installing a Cluster on vSphere with Customizations.

**NetApp Storage Overview**

NetApp has several storage platforms that are qualified with our Astra Trident Storage Orchestrator to provision storage for applications deployed on Red Hat OpenShift.

- AFF and FAS systems run NetApp ONTAP and provide storage for both file-based (NFS) and block-based (iSCSI) use cases.
- Cloud Volumes ONTAP and ONTAP Select provide the same benefits in the cloud and virtual space respectively.
- NetApp Cloud Volumes Service (AWS/GCP) and Azure NetApp Files provide file-based storage in the cloud.
- NetApp Element storage systems provide for block-based (iSCSI) use cases in a highly scalable environment.

Each storage system in the NetApp portfolio can ease both data management and movement between on-premises sites and the cloud, ensuring that your data is where your applications are.

The following pages have additional information about the NetApp storage systems validated in the Red Hat OpenShift with NetApp solution:

- NetApp ONTAP
- NetApp Element

**NetApp ONTAP**

NetApp ONTAP is a powerful storage-software tool with capabilities such as an intuitive GUI, REST APIs with
automation integration, AI-informed predictive analytics and corrective action, non-disruptive hardware upgrades, and cross-storage import.

For more information about the NetApp ONTAP storage system, visit the NetApp ONTAP website.

ONTAP provides the following features:

- A unified storage system with simultaneous data access and management of NFS, CIFS, iSCSI, FC, FCoE, and FC-NVMe protocols.
- Different deployment models include on-premises on all-flash, hybrid, and all-HDD hardware configurations; VM-based storage platforms on a supported hypervisor such as ONTAP Select; and in the cloud as Cloud Volumes ONTAP.
- Increased data storage efficiency on ONTAP systems with support for automatic data tiering, inline data compression, deduplication, and compaction.
- Workload-based, QoS-controlled storage.
- Seamless integration with a public cloud for tiering and protection of data. ONTAP also provides robust data protection capabilities that sets it apart in any environment:
  - **NetApp Snapshot copies.** A fast, point-in-time backup of data using a minimal amount of disk space with no additional performance overhead.
  - **NetApp SnapMirror.** Mirrors the Snapshot copies of data from one storage system to another. ONTAP supports mirroring data to other physical platforms and cloud-native services as well.
  - **NetApp SnapLock.** Efficiently administration of non-rewritable data by writing it to special volumes that cannot be overwritten or erased for a designated period.
  - **NetApp SnapVault.** Backs up data from multiple storage systems to a central Snapshot copy that serves as a backup to all designated systems.
  - **NetApp SyncMirror.** Provides real-time, RAID-level mirroring of data to two different plexes of disks that are connected physically to the same controller.
  - **NetApp SnapRestore.** Provides fast restoration of backed-up data on demand from Snapshot copies.
  - **NetApp FlexClone.** Provides instantaneous provisioning of a fully readable and writeable copy of a NetApp volume based on a Snapshot copy.

For more information about ONTAP, see the ONTAP 9 Documentation Center.

NetApp ONTAP is available on-premises, virtualized, or in the cloud.
NetApp platforms

**NetApp AFF/FAS**

NetApp provides robust all-flash (AFF) and scale-out hybrid (FAS) storage platforms that are tailor-made with low-latency performance, integrated data protection, and multi-protocol support.

Both systems are powered by NetApp ONTAP data management software, the industry’s most advanced data-management software for highly-available, cloud-integrated, simplified storage management to deliver enterprise-class speed, efficiency, and security for your data fabric needs.

For more information about NETAPP AFF/FAS platforms, click here.

**ONTAP Select**

ONTAP Select is a software-defined deployment of NetApp ONTAP that can be deployed onto a hypervisor in your environment. It can be installed on VMware vSphere or on KVM and provides the full functionality and experience of a hardware-based ONTAP system.

For more information about ONTAP Select, click here.

**Cloud Volumes ONTAP**

NetApp Cloud Volumes ONTAP is a cloud-deployed version of NetApp ONTAP available to be deployed in a number of public clouds, including: Amazon AWS, Microsoft Azure, and Google Cloud.
For more information about Cloud Volumes ONTAP, click here.

NetApp Element: Red Hat OpenShift with NetApp

NetApp Element software provides modular, scalable performance, with each storage node delivering guaranteed capacity and throughput to the environment. NetApp Element systems can scale from 4 to 100 nodes in a single cluster and offer a number of advanced storage management features.

For more information about NetApp Element storage systems, visit the NetApp Solidfire website.

iSCSI login redirection and self-healing capabilities

NetApp Element software leverages the iSCSI storage protocol, a standard way to encapsulate SCSI commands on a traditional TCP/IP network. When SCSI standards change or when the performance of Ethernet networks improves, the iSCSI storage protocol benefits without the need for any changes.

Although all storage nodes have a management IP and a storage IP, NetApp Element software advertises a single storage virtual IP address (SVIP address) for all storage traffic in the cluster. As a part of the iSCSI login process, storage can respond that the target volume has been moved to a different address and therefore it cannot proceed with the negotiation process. The host then reissues the login request to the new address in a process that requires no host-side reconfiguration. This process is known as iSCSI login redirection.

iSCSI login redirection is a key part of the NetApp Element software cluster. When a host login request is received, the node decides which member of the cluster should handle the traffic based on the IOPS and the capacity requirements for the volume. Volumes are distributed across the NetApp Element software cluster and are redistributed if a single node is handling too much traffic for its volumes or if a new node is added. Multiple copies of a given volume are allocated across the array.

In this manner, if a node failure is followed by volume redistribution, there is no effect on host connectivity beyond a logout and login with redirection to the new location. With iSCSI login redirection, a NetApp Element software cluster is a self-healing, scale-out architecture that is capable of non-disruptive upgrades and operations.

NetApp Element software cluster QoS

A NetApp Element software cluster allows QoS to be dynamically configured on a per-volume basis. You can use per-volume QoS settings to control storage performance based on SLAs that you define. The following three configurable parameters define the QoS:

- **Minimum IOPS.** The minimum number of sustained IOPS that the NetApp Element software cluster provides to a volume. The minimum IOPS configured for a volume is the guaranteed level of performance for a volume. Per-volume performance does not drop below this level.
- **Maximum IOPS.** The maximum number of sustained IOPS that the NetApp Element software cluster provides to a particular volume.
• **Burst IOPS.** The maximum number of IOPS allowed in a short burst scenario. The burst duration setting is configurable, with a default of 1 minute. If a volume has been running below the maximum IOPS level, burst credits are accumulated. When performance levels become very high and are pushed, short bursts of IOPS beyond the maximum IOPS are allowed on the volume.

**Multitenancy**

Multitenancy is achieved with the following features:

- **Secure authentication.** The Challenge-Handshake Authentication Protocol (CHAP) is used for secure volume access. The Lightweight Directory Access Protocol (LDAP) is used for secure access to the cluster for management and reporting.

- **Volume access groups (VAGs).** Optionally, VAGs can be used in lieu of authentication, mapping any number of iSCSI initiator-specific iSCSI Qualified Names (IQNs) to one or more volumes. To access a volume in a VAG, the initiator’s IQN must be in the allowed IQN list for the group of volumes.

- **Tenant virtual LANs (VLANs).** At the network level, end-to-end network security between iSCSI initiators and the NetApp Element software cluster is facilitated by using VLANs. For any VLAN that is created to isolate a workload or a tenant, NetApp Element Software creates a separate iSCSI target SVIP address that is accessible only through the specific VLAN.

- **VRF-enabled VLANs.** To further support security and scalability in the data center, NetApp Element software allows you to enable any tenant VLAN for VRF-like functionality. This feature adds these two key capabilities:
  - **L3 routing to a tenant SVIP address.** This feature allows you to situate iSCSI initiators on a separate network or VLAN from that of the NetApp Element software cluster.
  - **Overlapping or duplicate IP subnets.** This feature enables you to add a template to tenant environments, allowing each respective tenant VLAN to be assigned IP addresses from the same IP subnet. This capability can be useful for in-service provider environments where scale and preservation of IP space are important.

**Enterprise storage efficiencies**

The NetApp Element software cluster increases overall storage efficiency and performance. The following features are performed inline, are always on, and require no manual configuration by the user:

- **Deduplication.** The system only stores unique 4K blocks. Any duplicate 4K blocks are automatically associated to an already stored version of the data. Data is on block drives and is mirrored by using the NetApp Element software Helix data protection. This system significantly reduces capacity consumption and write operations within the system.

- **Compression.** Compression is performed inline before data is written to NVRAM. Data is compressed, stored in 4K blocks, and remains compressed in the system. This compression significantly reduces capacity consumption, write operations, and bandwidth consumption across the cluster.

- **Thin-provisioning.** This capability provides the right amount of storage at the time that you need it, eliminating capacity consumption that caused by overprovisioned volumes or underutilized volumes.

- **Helix.** The metadata for an individual volume is stored on a metadata drive and is replicated to a secondary metadata drive for redundancy.

Element was designed for automation. All the storage features are available through APIs. These APIs are the only method that the UI uses to control the system.
NetApp Storage Integration Overview

NetApp provides a number of products to help you with orchestrating and managing persistent data in container based environments, such as Red Hat OpenShift.

NetApp Astra Control offers a rich set of storage and application-aware data management services for stateful Kubernetes workloads, powered by NetApp data protection technology. The Astra Control Service is available to support stateful workloads in cloud-native Kubernetes deployments. The Astra Control Center is available to support stateful workloads in on-premises deployments, like Red Hat OpenShift. For more information visit the NetApp Astra Control website here.

NetApp Astra Trident is an open-source and fully-supported storage orchestrator for containers and Kubernetes distributions, including Red Hat OpenShift. For more information, visit the Astra Trident website here.

The following pages have additional information about the NetApp products that have been validated for application and persistent storage management in the Red Hat OpenShift with NetApp solution:

- NetApp Astra Control Center
NetApp Astra Control Center overview

NetApp Astra Control Center offers a rich set of storage and application-aware data management services for stateful Kubernetes workloads deployed in an on-premises environment and powered by NetApp data protection technology.

NetApp Astra Control Center can be installed on a Red Hat OpenShift cluster that has the Astra Trident storage orchestrator deployed and configured with storage classes and storage backends to NetApp ONTAP storage systems.

For the installation and configuration of Astra Trident to support Astra Control Center, see this document here.

In a cloud-connected environment, Astra Control Center uses Cloud Insights to provide advanced monitoring and telemetry. In the absence of a Cloud Insights connection, limited monitoring and telemetry (7-days worth of metrics) is available and exported to Kubernetes native monitoring tools (Prometheus and Grafana) through open metrics endpoints.

Astra Control Center is fully integrated into the NetApp AutoSupport and Active IQ ecosystem to provide support for users, provide assistance with troubleshooting, and display usage statistics.

In addition to the paid version of Astra Control Center, a 90-day evaluation license is available. The evaluation version is supported through the email and community (Slack channel). Customers have access to these and other knowledge-base articles and the documentation available from the in-product support dashboard.

To get started with NetApp Astra Control Center, visit the Astra website.
Astra Control Center installation prerequisites

1. One or more Red Hat OpenShift clusters. Versions 4.6 EUS and 4.7 are currently supported.
2. Astra Trident must already be installed and configured on each Red Hat OpenShift cluster.
3. One or more NetApp ONTAP storage systems running ONTAP 9.5 or greater.
   - It's best practice for each OpenShift install at a site to have a dedicated SVM for persistent storage. Multi-site deployments require additional storage systems.
4. A Trident storage backend must be configured on each OpenShift cluster with an SVM backed by an ONTAP cluster.
5. A default StorageClass configured on each OpenShift cluster with Astra Trident as the storage provisioner.
6. A load balancer must be installed and configured on each OpenShift cluster for load balancing and exposing OpenShift Services.
   - See the link here for information about load balancers that have been validated for this purpose.
7. A private image registry must be configured to host the NetApp Astra Control Center images.
   - See the link here to install and configure an OpenShift private registry for this purpose.
8. You must have Cluster Admin access to the Red Hat OpenShift cluster.
9. You must have Admin access to NetApp ONTAP clusters.
10. An admin workstation with docker or podman, tridentctl, and oc or kubectl tools installed and added to your $PATH.
    - Docker installations must have docker version greater than 20.10 and Podman installations must have podman version greater than 3.0.

Install Astra Control Center
Using OperatorHub

1. Log into the NetApp Support Site and download the latest version of NetApp Astra Control Center. To do so requires a license attached to your NetApp account. After you download the tarball, transfer it to the admin workstation.

   To get started with a trial license for Astra Control, visit the Astra registration site.

2. Unpack the tar ball and change the working directory to the resulting folder.

   ```
   $ tar -vxzf astra-control-center-21.12.60.tar.gz
   $ cd astra-control-center-21.12.60
   ```

3. Before starting the installation, push the Astra Control Center images to an image registry. You can choose to do this with either Docker or Podman, instructions for both are provided in this step.
Podman

a. Export the registry FQDN with the organization/namespace/project name as an environment variable ‘registry’.

[netapp-user@rhel7 ~]$ export REGISTRY=astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra

b. Log into the registry.

[netapp-user@rhel7 ~]$ podman login -u ocp-user -p password --tls-verify=false astra-registry.apps.ocp-vmw.cie.netapp.com

If you are using kubeadmin user to log into the private registry, then use token instead of password - podman login -u ocp-user -p token --tls-verify=false astra-registry.apps.ocp-vmw.cie.netapp.com.

Alternatively, you can create a service account, assign registry-editor and/or registry-viewer role (based on whether you require push/pull access) and log into the registry using service account’s token.

c. Create a shell script file and paste the following content in it.

[netapp-user@rhel7 ~]$ vi push-images-to-registry.sh

for astraImageFile in $(ls images/*.tar) ; do
  # Load to local cache. And store the name of the loaded image trimming the 'Loaded images: '
  astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded image(s): //')
  astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
  # Tag with local image repo.
  podman tag ${astraImage} ${REGISTRY}/${astraImage}
  # Push to the local repo.
  podman push ${REGISTRY}/${astraImage}
done

If you are using untrusted certificates for your registry, edit the shell script and use --tls-verify=false for the podman push command podman push $REGISTRY/$(echo $astraImage | sed 's/\[//\]++\///') --tls-verify=false.

d. Make the file executable.
e. Execute the shell script.

[netapp-user@rhel7 ~]$ ./push-images-to-registry.sh
Docker

a. Export the registry FQDN with the organization/namespace/project name as a environment variable ‘registry’.

```
[netapp-user@rhel7 ~]$ export REGISTRY=astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra
```

b. Log into the registry.

```
[netapp-user@rhel7 ~]$ docker login -u ocp-user -p password astra-registry.apps.ocp-vmw.cie.netapp.com
```

If you are using `kubeadmin` user to log into the private registry, then use token instead of password - docker login -u ocp-user -p token astra-registry.apps.ocp-vmw.cie.netapp.com.

Alternatively, you can create a service account, assign registry-editor and/or registry-viewer role (based on whether you require push/pull access) and log into the registry using service account’s token.

c. Create a shell script file and paste the following content in it.

```
[netapp-user@rhel7 ~]$ vi push-images-to-registry.sh

for astraImageFile in $(ls images/*.tar) ; do
    # Load to local cache. And store the name of the loaded image trimming the 'Loaded images: '
    astraImage=$(docker load --input ${astraImageFile} | sed 's/Loaded image: //')
    astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
    # Tag with local image repo.
    docker tag ${astraImage} ${REGISTRY}/${astraImage}
    # Push to the local repo.
    docker push ${REGISTRY}/${astraImage}
done
```

d. Make the file executable.

```
[netapp-user@rhel7 ~]$ chmod +x push-images-to-registry.sh
```

e. Execute the shell script.
4. When using private image registries that are not publicly trusted, upload the image registry TLS certificates to the OpenShift nodes. To do so, create a configmap in the openshift-config namespace using the TLS certificates and patch it to the cluster image config to make the certificate trusted.

```
[netapp-user@rhel7 ~]$ ./push-images-to-registry.sh
```

```
[netapp-user@rhel7 ~]$ oc create configmap default-ingress-ca -n openshift-config --from-file=astra-registry.apps.ocp -vmw.cie.netapp.com=tls.crt
```

```
[netapp-user@rhel7 ~]$ oc patch image.config.openshift.io/cluster --patch '{"spec":{"additionalTrustedCA":{"name":"default-ingress-ca"}}}'} --type=merge
```

```
If you are using an OpenShift internal registry with default TLS certificates from the ingress operator with a route, you still need to follow the previous step to patch the certificates to the route hostname. To extract the certificates from ingress operator, you can use the command:
```
```
oc extract secret/router-ca --keys=tls.crt -n openshift-ingress-operator
```
```

5. Create a namespace `netapp-acc-operator` for Astra Control Center.

```
[netapp-user@rhel7 ~]$ oc create ns netapp-acc-operator
```

```
namespace/netapp-acc-operator created
```

6. Create a secret with credentials to log into the image registry in `netapp-acc-operator` namespace.

```
[netapp-user@rhel7 ~]$ oc create secret docker-registry astra-registry-cred --docker-server=astra-registry.apps.ocp -vmw.cie.netapp.com --docker-username=ocp-user --docker-password=password -n netapp-acc-operator
```

```
secret/astra-registry-cred created
```

7. Log into the Red Hat OpenShift GUI console with cluster-admin access.

8. Select Administrator from the Perspective drop down.

10. Select `netapp-acc-operator` tile and click Install.

**netapp-acc-operator**

2112.63-1 provided by NetApp

- **Latest version**: 2112.63-1
- **Capability level**: Basic Install
  - Seamless Upgrades
  - Full Lifecycle
  - Deep Insights
  - Auto Pilot
- **Provider type**: Certified
- **Provider**: NetApp

Astra Control is an application-aware data management solution that manages, protects and moves data-rich Kubernetes workloads in both public clouds and on-premises.

Astra Control enables data protection, disaster recovery, and migration for your Kubernetes workloads, leveraging NetApp’s industry-leading data management technology for snapshots, backups, replication and cloning.

**How to deploy Astra Control**

Refer to [Installation Procedure](#) to deploy Astra Control Center using the Operator.

**Documentation**

Refer to [Astra Control Center Documentation](#) to complete the setup and start managing applications.

11. On the Install Operator screen, accept all default parameters and click Install.
Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Update channel *
- alpha
- stable

Installation mode *
- All namespaces on the cluster (default) 
  Operator will be available in all Namespaces.
- A specific namespace on the cluster 
  This mode is not supported by this Operator.

Installed Namespace *
- netapp-acc-operator (Operator recommended)

```
Namespace already exists

Namespace netapp-acc-operator already exists and will be used. Other users can already have access to this namespace.
```

Approval strategy *
- Automatic
- Manual

12. Wait for the operator installation to complete.

```
netapp-acc-operator
2112.63-I provided by NetApp

Installing Operator

InstallWaiting: installing: waiting for deployment acc-operator-controller-manager to become ready: Waiting for rollout to finish: 0 of 1 updated replicas are available...

The Operator is being installed. This may take a few minutes.

View installed Operators in Namespace netapp-acc-operator
```

13. Once the operator installation succeeds, navigate to click on View Operator.
14. Then click on Create Instance in Astra Control Center tile in the operator.

15. Fill the Create AstraControlCenter form fields and click Create.
   a. Optionally edit the Astra Control Center instance name.
   b. Optionally enable or disable Auto Support. Retaining Auto Support functionality is recommended.
   c. Enter the FQDN for Astra Control Center.
   d. Enter the Astra Control Center version; the latest is displayed by default.
   e. Enter an account name for Astra Control Center and admin details like first name, last name and
email address.

f. Enter the volume reclaim policy, default is Retain.

g. In Image Registry, enter the FQDN for your registry along with the organization name as it was given while pushing the images to the registry (in this example, astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra)

h. If you use a registry that requires authentication, enter the secret name in Image Registry section.

i. Configure scaling options for Astra Control Center resource limits.

j. Enter the storage class name if you want to place PVCs on a non-default storage class.

k. Define CRD handling preferences.

Project: netapp-acc-operator

Name *

astra

Labels

app=frontend

Account Name *

HCG Solutions Engineering

Astra Control Center account name

Astra Address *

astra-control-center.cie.netapp.com

AstraAddress defines how Astra will be found in the data center. This IP address and/or DNS A record must be created prior to provisioning Astra Control Center. Example - “astra.example.com” The A record and its IP address must be allocated prior to provisioning Astra Control Center

Astra Version *

21.2.60

Version of AstraControlCenter to deploy. You are provided a Helm repository with a corresponding version. Example – 1.5.2, 1.4.2-patch

Email *

solutions_tme@netapp.com

EmailAddress will be notified by Astra as events warrant.

Auto Support *

AutoSupport indicates willingness to participate in NetApp's proactive support application, NetApp Active IQ. The default election is true and indicates support data will be sent to NetApp. An empty or blank election is the same as a default election. Air gapped installations should enter false.

First Name

HCG

The first name of the SRE supporting Astra.
Automated [Ansible]

1. To use Ansible playbooks to deploy Astra Control Center, you need an Ubuntu/RHEL machine with Ansible installed. Follow the procedures [here](#) for Ubuntu and RHEL.

2. Clone the GitHub repository that hosts the Ansible content.

   ```
git clone https://github.com/NetApp-Automation/na_astra_control_suite.git
   ```

3. Log into the NetApp Support site and download the latest version of NetApp Astra Control Center. To do so requires a license attached to your NetApp account. After you download the tarball, transfer it to the workstation.

   To get started with a trial license for Astra Control, visit the [Astra registration site](#).

4. Create or obtain the kubeconfig file with admin access to the OpenShift cluster on which Astra Control Center is to be installed.

5. Change the directory to the `na_astra_control_suite`.
6. Edit the `vars/vars.yml` file, and fill in the variables with the required information.

```yaml
# Define whether or not to push the Astra Control Center images to your private registry [Allowed values: yes, no]
push_images: yes

# The directory hosting the Astra Control Center installer
installer_directory: /home/admin/

# Specify the ingress type. Allowed values - "AccTraefik" or "Generic"
# "AccTraefik" if you want the installer to create a LoadBalancer type service to access ACC, requires MetalLB or similar.
# "Generic" if you want to create or configure ingress controller yourself, installer just creates a ClusterIP service for traefik.
ingress_type: "AccTraefik"

# Name of the Astra Control Center installer (Do not include the extension, just the name)
astra_tar_ball_name: astra-control-center-22.04.0

# The complete path to the kubeconfig file of the kubernetes/openshift cluster Astra Control Center needs to be installed to.
hosting_k8s_cluster_kubeconfig_path: /home/admin/cluster-kubeconfig.yml

# Namespace in which Astra Control Center is to be installed
astra_namespace: netapp-astra-cc

# Astra Control Center Resources Scaler. Leave it blank if you want to accept the Default setting.
astra_resources_scaler: Default

# Storageclass to be used for Astra Control Center PVCs, it must be created before running the playbook [Leave it blank if you want the PVCs to use default storageclass]
astra_trident_storageclass: basic

# Reclaim Policy for Astra Control Center Persistent Volumes [Allowed values: Retain, Delete]
storageclass_reclaim_policy: Retain
```
# Private Registry Details
astra_registry_name: "docker.io"

# Whether the private registry requires credentials [Allowed values: yes, no]
require_reg_creds: yes

# If require_reg_creds is yes, then define the container image registry credentials
# Usually, the registry namespace and usernames are same for individual users
astra_registry_namespace: "registry-user"
astra_registry_username: "registry-user"
astra_registry_password: "password"

# Kubernetes/OpenShift secret name for Astra Control Center
# This name will be assigned to the K8s secret created by the playbook
astra_registry_secret_name: "astra-registry-credentials"

# Astra Control Center FQDN
acc_fqdn_address: astra-control-center.cie.netapp.com

# Name of the Astra Control Center instance
acc_account_name: ACC Account Name

# Administrator details for Astra Control Center
admin_email_address: admin@example.com
admin_first_name: Admin
admin_last_name: Admin

7. Run the playbook to deploy Astra Control Center. The playbook requires root privileges for certain configurations.

   If the user running the playbook is root or has passwordless sudo configured, then run the following command to run the playbook.


```bash
ansible-playbook install_acc_playbook.yml
```

   If the user has password-based sudo access configured, run the following command to run the playbook, and then enter the sudo password.


```bash
ansible-playbook install_acc_playbook.yml -K
```
Post Install Steps

1. It might take several minutes for the installation to complete. Verify that all the pods and services in the netapp-astra-cc namespace are up and running.

   [netapp-user@rhel7 ~]$ oc get all -n netapp-astra-cc

2. Check the acc-operator-controller-manager logs to ensure that the installation is completed.

   [netapp-user@rhel7 ~]$ oc logs deploy/acc-operator-controller-manager -n netapp-acc-operator -c manager -f

   The following message indicates the successful installation of Astra Control Center.

   ```json
   {"level":"info","ts":1624054318.029971,"logger":"controllers.AstraControlCenter","msg":"Successfully Reconciled AstraControlCenter in [seconds]s","AstraControlCenter":"netapp-astra-cc/astra","ae.Version":"[21.12.60]"}
   ```

3. The username for logging into Astra Control Center is the email address of the administrator provided in the CRD file and the password is a string ACC- appended to the Astra Control Center UUID. Run the following command:

   [netapp-user@rhel7 ~]$ oc get astracontrolcenters -n netapp-astra-cc

<table>
<thead>
<tr>
<th>NAME</th>
<th>UUID</th>
</tr>
</thead>
<tbody>
<tr>
<td>astra</td>
<td>345c55a5-bf2e-21f0-84b8-b6f2bce5e95f</td>
</tr>
</tbody>
</table>

   In this example, the password is ACC-345c55a5-bf2e-21f0-84b8-b6f2bce5e95f.

4. Get the traefik service load balancer IP.

   [netapp-user@rhel7 ~]$ oc get svc -n netapp-astra-cc | egrep 'EXTERNAL|traefik'

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLUSTER-IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>traefik</td>
<td>LoadBalancer</td>
<td>172.30.99.142</td>
</tr>
<tr>
<td>10.61.186.181</td>
<td>80:30343/TCP,443:30060/TCP</td>
<td></td>
</tr>
<tr>
<td>16m</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Add an entry in the DNS server pointing the FQDN provided in the Astra Control Center CRD file to the
6. Log into the Astra Control Center GUI by browsing its FQDN.
7. When you log into Astra Control Center GUI for the first time using the admin email address provided in CRD, you need to change the password.

8. If you wish to add a user to Astra Control Center, navigate to Account > Users, click Add, enter the details of the user, and click Add.
9. Astra Control Center requires a license for all of its functionalities to work. To add a license, navigate to Account > License, click Add License, and upload the license file.

If you encounter issues with the install or configuration of NetApp Astra Control Center, the knowledge base of known issues is available here.

Register your Red Hat OpenShift Clusters with the Astra Control Center

To enable the Astra Control Center to manage your workloads, you must first register your Red Hat OpenShift cluster.

Register Red Hat OpenShift clusters

1. The first step is to add the OpenShift clusters to the Astra Control Center and manage them. Go to Clusters
and click Add a Cluster, upload the kubeconfig file for the OpenShift cluster, and click Select Storage.

The kubeconfig file can be generated to authenticate with a username and password or a token. Tokens expire after a limited amount of time and might leave the registered cluster unreachable. NetApp recommends using a kubeconfig file with a username and password to register your OpenShift clusters to Astra Control Center.

2. Astra Control Center detects the eligible storage classes. Now select the way that storageclass provisions volumes using Trident backed by an SVM on NetApp ONTAP and click Review. In the next pane, verify the details and click Add Cluster.
3. Register both OpenShift clusters as described in step 1. When added, the clusters move to the Discovering status while Astra Control Center inspects them and installs the necessary agents. Cluster status changes to Running after they are successfully registered.

All Red Hat OpenShift clusters to be managed by Astra Control Center should have access to the image registry that was used for its installation as the agents installed on the managed clusters pull the images from that registry.

4. Import ONTAP clusters as storage resources to be managed as backends by Astra Control Center. When OpenShift clusters are added to Astra and a storageclass is configured, it automatically discovers and inspects the ONTAP cluster backing the storageclass but does not import it into the Astra Control Center to be managed.
5. To import the ONTAP clusters, go to Backends, click the dropdown, and select Manage next to the ONTAP cluster to be managed. Enter the ONTAP cluster credentials, click Review Information, and then click Import Storage Backend.

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Capacity</th>
<th>Type</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONTAP</td>
<td>Not available yet</td>
<td></td>
<td>ONTAP</td>
<td></td>
</tr>
<tr>
<td>ONTAP</td>
<td>Not available yet</td>
<td></td>
<td>ONTAP</td>
<td></td>
</tr>
</tbody>
</table>

6. After the backends are added, the status changes to Available. These backends now have the information about the persistent volumes in the OpenShift cluster and the corresponding volumes on the ONTAP system.
7. For backup and restore across OpenShift clusters using Astra Control Center, you must provision an object storage bucket that supports the S3 protocol. Currently supported options are ONTAP S3, StorageGRID, and AWS S3. For the purpose of this installation, we are going to configure an AWS S3 bucket. Go to Buckets, click Add bucket, and select Generic S3. Enter the details about the S3 bucket and credentials to access it, click the checkbox "Make this bucket the default bucket for the cloud," and then click Add.

Choose the applications to protect

After you have registered your Red Hat OpenShift clusters, you can discover the applications that are deployed and manage them via the Astra Control Center.
Manage applications

1. After the OpenShift clusters and ONTAP backends are registered with the Astra Control Center, the control center automatically starts discovering the applications in all the namespaces that are using the storageclass configured with the specified ONTAP backend.

2. Navigate to Apps > Discovered and click the dropdown menu next to the application you would like to manage using Astra. Then click Manage.

1. The application enters the Available state and can be viewed under the Managed tab in the Apps section.
Protect your applications

After application workloads are managed by Astra Control Center, you can configure the protection settings for those workloads.

Creating an application snapshot

A snapshot of an application creates an ONTAP Snapshot copy that can be used to restore or clone the application to a specific point in time based on that Snapshot copy.

1. To take a snapshot of the application, navigate to the Apps > Managed tab and click the application you would like to make a Snapshot copy of. Click the dropdown menu next to the application name and click Snapshot.

2. Enter the snapshot details, click Next, and then click Snapshot. It takes about a minute to create the snapshot, and the status becomes Available after the snapshot is successfully created.
Creating an application backup

A backup of an application captures the active state of the application and the configuration of its resources, converts them into files, and stores them in a remote object storage bucket.

For the backup and restore of managed applications in the Astra Control Center, you must configure superuser settings for the backing ONTAP systems as a prerequisite. To do so, enter the following commands.

```
ONTAP::> export-policy rule modify -vserver ocp-trident -policyname default -ruleindex 1 -superuser sys
ONTAP::> export-policy rule modify -policyname default -ruleindex 1 -anon 65534 -vserver ocp-trident
```

1. To create a backup of the managed application in the Astra Control Center, navigate to the Apps > Managed tab and click the application that you want to take a backup of. Click the dropdown menu next to the application name and click Backup.

2. Enter the backup details, select the object storage bucket to hold the backup files, click Next, and, after reviewing the details, click Backup. Depending on the size of the application and data, the backup can take several minutes, and the status of the backup becomes Available after the backup is completed successfully.
Restoring an application

At the push of a button, you can restore an application to the originating namespace in the same cluster or to a remote cluster for application protection and disaster recovery purposes.

1. To restore an application, navigate to Apps > Managed tab and click the app in question. Click the dropdown menu next to the application name and click Restore.

2. Enter the name of the restore namespace, select the cluster you want to restore it to, and choose if you want to restore it from an existing snapshot or from a backup of the application. Click Next.
3. On the review pane, enter restore and click Restore after you have reviewed the details.

4. The new application goes to the Restoring state while Astra Control Center restores the application on the selected cluster. After all the resources of the application are installed and detected by Astra, the application goes to the Available state.
Cloning an application

You can clone an application to the originating cluster or to a remote cluster for dev/test or application protection and disaster recovery purposes. Cloning an application within the same cluster on the same storage backend uses NetApp FlexClone technology, which clones the PVCs instantly and saves storage space.

1. To clone an application, navigate to the Apps > Managed tab and click the app in question. Click the dropdown menu next to the application name and click Clone.

2. Enter the details of the new namespace, select the cluster you want to clone it to, and choose if you want to clone it from an existing snapshot or a backup or the current state of the application. Then click Next and click Clone on review pane once you have reviewed the details.

3. The new application goes to the Discovering state while Astra Control Center creates the application on the
selected cluster. After all the resources of the application are installed and detected by Astra, the application goes to the Available state.

Astra Trident Overview

Astra Trident is an open-source and fully supported storage orchestrator for containers and Kubernetes distributions, including Red Hat OpenShift. Trident works with the entire NetApp storage portfolio, including the NetApp ONTAP and Element storage systems, and it also supports NFS and iSCSI connections. Trident accelerates the DevOps workflow by allowing end users to provision and manage storage from their NetApp storage systems without requiring intervention from a storage administrator.

An administrator can configure a number of storage backends based on project needs and storage system models that enable advanced storage features, including compression, specific disk types, or QoS levels that guarantee a certain level of performance. After they are defined, these backends can be used by developers in their projects to create persistent volume claims (PVCs) and to attach persistent storage to their containers on demand.

Astra Trident has a rapid development cycle, and just like Kubernetes, is released four times a year.

The latest version of Astra Trident is 22.01 released in January 2022. A support matrix for what version of Trident has been tested with which Kubernetes distribution can be found here.
Starting with the 20.04 release, Trident setup is performed by the Trident operator. The operator makes large scale deployments easier and provides additional support including self healing for pods that are deployed as a part of the Trident install.

With the 21.01 release, a Helm chart was made available to ease the installation of the Trident Operator.

**Download Astra Trident**

To install Trident on the deployed user cluster and provision a persistent volume, complete the following steps:

1. Download the installation archive to the admin workstation and extract the contents. The current version of Trident is 22.01, which can be downloaded here.

```
[netapp-user@rhel7 ~]$ wget https://github.com/NetApp/trident/releases/download/v22.01.0/trident-installer-22.01.0.tar.gz
--2021-05-06 15:17:30--
https://github.com/NetApp/trident/releases/download/v22.01.0/trident-installer-22.01.0.tar.gz
Resolving github.com (github.com)... 140.82.114.3
Connecting to github.com (github.com)|140.82.114.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-releases.githubusercontent.com/77179634/a4fa9f00-a9f2-11eb-9053-98e8e573d4ae?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20210506%2Fs3%2Faws4_request&X-Amz-Date=20210506T191643Z&X-Amz-Expires=300&X-Amz-Signature=8a49a2a1e08c147d1dd8149ce45a5714f9853fee19bb1c507989b9543eb3630&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=77179634&response-content-disposition=attachment%3B%20filename%3Dtrident-installer-22.01.0.tar.gz&response-content-type=application%2Foctet-stream [following]
--2021-05-06 15:17:30--  https://github-releases.githubusercontent.com/77179634/a4fa9f00-a9f2-11eb-9053-98e8e573d4ae?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20210506%2Fs3%2Faws4_request&X-Amz-Date=20210506T191643Z&X-Amz-Expires=300&X-Amz-Signature=8a49a2a1e08c147d1dd8149ce45a5714f9853fee19bb1c507989b9543eb3630&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=77179634&response-content-disposition=attachment%3B%20filename%3Dtrident-installer-22.01.0.tar.gz&response-content-type=application%2Foctet-stream
Resolving github-releases.githubusercontent.com (github-releases.githubusercontent.com)... 185.199.108.154, 185.199.109.154, 185.199.110.154, ...
```
2. Extract the Trident install from the downloaded bundle.

```
[netapp-user@rhel7 ~]$ tar -xzf trident-installer-22.01.0.tar.gz
[netapp-user@rhel7 ~]$ cd trident-installer/
[netapp-user@rhel7 trident-installer]$ 
```

Install the Trident Operator with Helm

1. First set the location of the user cluster's `kubeconfig` file as an environment variable so that you don't have to reference it, because Trident has no option to pass this file.

```
[netapp-user@rhel7 trident-installer]$ export KUBECONFIG=~/.ocp-install/auth/kubeconfig
```

2. Run the Helm command to install the Trident operator from the tarball in the helm directory while creating the trident namespace in your user cluster.
Thank you for installing trident-operator, which will deploy and manage NetApp's Trident CSI storage provisioner for Kubernetes.

Your release is named 'trident' and is installed into the 'trident' namespace.
Please note that there must be only one instance of Trident (and trident-operator) in a Kubernetes cluster.

To configure Trident to manage storage resources, you will need a copy of tridentctl, which is available in pre-packaged Trident releases. You may find all Trident releases and source code online at https://github.com/NetApp/trident.

To learn more about the release, try:

$ helm status trident
$ helm get all trident

3. You can verify that Trident is successfully installed by checking the pods that are running in the namespace or by using the tridentctl binary to check the installed version.
In some cases, customer environments might require the customization of the Trident deployment. In these cases, it is also possible to manually install the Trident operator and update the included manifests to customize the deployment.

**Manually install the Trident Operator**

1. First, set the location of the user cluster’s `kubeconfig` file as an environment variable so that you don’t have to reference it, because Trident has no option to pass this file.

   ```
   [netapp-user@rhel7 trident-installer]$ export KUBECONFIG=~/.ocp-install/auth/kubeconfig
   ```

2. The `trident-installer` directory contains manifests for defining all the required resources. Using the appropriate manifests, create the `TridentOrchestrator` custom resource definition.

   ```
   [netapp-user@rhel7 trident-installer]$ oc create -f deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
   customresourcedefinition.apiextensions.k8s.io/tridentorchestrators.trident.netapp.io created
   ```

3. If one does not exist, create a Trident namespace in your cluster using the provided manifest.

   ```
   [netapp-user@rhel7 trident-installer]$ oc apply -f deploy/namespace.yaml
   namespace/trident created
   ```

4. Create the resources required for the Trident operator deployment, such as a ServiceAccount for the operator, a ClusterRole and ClusterRoleBinding to the ServiceAccount, a dedicated PodSecurityPolicy, or the operator itself.
5. You can check the status of the operator after it's deployed with the following commands:

```
[netapp-user@rhel7 trident-installer]$ oc get deployment -n trident
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
trident-operator   1/1     1            1           23s
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-operator-66f48895cc-lzczk   1/1     Running   0          41s
```

6. With the operator deployed, we can now use it to install Trident. This requires creating a TridentOrchestrator.

```
[netapp-user@rhel7 trident-installer]$ oc create -f deploy/crds/tridentorchestrator_cr.yaml
tridentorchestrator.trident.netapp.io/trident created
[netapp-user@rhel7 trident-installer]$ oc describe torc trident
Name:         trident
Namespace:    <none>
Labels:       <none>
Annotations:  <none>
API Version:  trident.netapp.io/v1
Kind:         TridentOrchestrator
Metadata:     
Creation Timestamp:  2021-05-07T17:00:28Z
Generation:    1
Managed Fields:  
API Version:   trident.netapp.io/v1
Fields Type:   FieldsV1
fieldsV1:      
  f:spec:       
    .:          
    f:debug:    
    f:namespace:
Manager:       kubectl-create
Operation:     Update
Time:          2021-05-07T17:00:28Z
API Version:   trident.netapp.io/v1
```
Fields Type:  FieldsV1
fieldsV1:
  f:status:
   .:
   f:currentInstallationParams:
    .:
    f:IPv6:
    f:autosupportHostname:
    f:autosupportImage:
    f:autosupportProxy:
    f:autosupportSerialNumber:
    f:debug:
    f:enableNodePrep:
    f:imagePullSecrets:
    f:imageRegistry:
    f:k8sTimeout:
    f:kubeletDir:
    f:logFormat:
    f:silenceAutosupport:
    f:tridentImage:
    f:message:
    f:namespace:
    f:status:
    f:version:
Manager:   trident-operator
Operation:  Update
Time:      2021-05-07T17:00:28Z
Resource Version:  931421
Self Link:
/apis/trident.netapp.io/v1/tridentorchestrators/trident
UID:        8a26a7a6-dde8-4d55-9b66-a7126754d81f
Spec:
  Debug:  true
  Namespace:  trident
Status:
  Current Installation Params:
    IPv6:   false
    Autosupport Hostname:
    Autosupport Image:  netapp/trident-autosupport:21.01
    Autosupport Proxy:
    Autosupport Serial Number:
    Debug:  true
    Enable Node Prep:  false
    Image Pull Secrets:
    Image Registry:
    k8sTimeout:  30
Kubelet Dir: /var/lib/kubelet
Log Format: text
Silence Autosupport: false
Trident Image: netapp/trident:22.01.0
Message: Trident installed
Namespace: trident
Status: Installed
Version: v22.01.0

Events:

<table>
<thead>
<tr>
<th>Type</th>
<th>Reason</th>
<th>Age</th>
<th>From</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Installing</td>
<td>80s</td>
<td>trident-operator.netapp.io</td>
<td>Installing Trident installed</td>
</tr>
<tr>
<td>Normal</td>
<td>Installed</td>
<td>68s</td>
<td>trident-operator.netapp.io</td>
<td>Trident installed</td>
</tr>
</tbody>
</table>

7. You can verify that Trident is successfully installed by checking the pods that are running in the namespace or by using the tridentctl binary to check the installed version.

```
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-bb64c6cb4-lmd6h         6/6     Running   0          82s
trident-csi-gn59q                   2/2     Running   0          82s
trident-csi-m4szj                   2/2     Running   0          82s
trident-csi-sb9k9                   2/2     Running   0          82s
trident-operator-66f48895cc-lzczk   1/1     Running   0          2m39s

[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident version
+----------------+----------------+
| SERVER VERSION | CLIENT VERSION |
+----------------+----------------+
| 22.01.0          | 22.01.0          |
+----------------+----------------+
```

Prepare worker nodes for storage

NFS

Most Kubernetes distributions come with the packages and utilities to mount NFS backends installed by default, including Red Hat OpenShift.

However, for NFSv3, there is no mechanism to negotiate concurrency between the client and the server. Hence the maximum number of client-side sunrpc slot table entries must be manually synced with supported value on the server to ensure the best performance for the NFS connection without the server having to decrease the window size of the connection.

For ONTAP, the supported maximum number of sunrpc slot table entries is 128 i.e. ONTAP can serve 128
concurrent NFS requests at a time. However, by default, Red Hat CoreOS/Red Hat Enterprise Linux has maximum of 65,536 sunrpc slot table entries per connection. We need to set this value to 128 and this can be done using Machine Config Operator (MCO) in OpenShift.

To modify the maximum sunrpc slot table entries in OpenShift worker nodes, complete the following steps:

1. Log into the OCP web console and navigate to Compute > Machine Configs. Click Create Machine Config. Copy and paste the YAML file and click Create.

   ```yaml
   apiVersion: machineconfiguration.openshift.io/v1
   kind: MachineConfig
   metadata:
     name: 98-worker-nfs-rpc-slot-tables
     labels:
       machineconfiguration.openshift.io/role: worker
   spec:
     config:
       ignition:
         version: 3.2.0
       storage:
         files:
           - contents:
               source: data:text/plain;charset=utf-8;base64,b3B0aW9ucyBzdW5ycG9zcmFudGVkIHN0cmluZw==
               filesystem: root
               mode: 420
               path: /etc/modprobe.d/sunrpc.conf
   ```

2. After the MCO is created, the configuration needs to be applied on all worker nodes and rebooted one by one. The whole process takes approximately 20 to 30 minutes. Verify whether the machine config is applied by using `oc get mcp` and make sure that the machine config pool for workers is updated.

   ```bash
   [netapp-user@rhe17 openshift-deploy]$ oc get mcp
   NAME           CONFIG                             UPDATED   UPDATING     DEGRADED
   master         rendered-master-a520ae930e1d135e0dee7168 True False
   worker         rendered-worker-de321b36eeba62df41feb7bc True False
   ```

**iSCSI**

To prepare worker nodes to allow for the mapping of block storage volumes through the iSCSI protocol, you must install the necessary packages to support that functionality.
In Red Hat OpenShift, this is handled by applying an MCO (Machine Config Operator) to your cluster after it is deployed.

To configure the worker nodes to run iSCSI services, complete the following steps:

1. Log into the OCP web console and navigate to Compute > Machine Configs. Click Create Machine Config. Copy and paste the YAML file and click Create.

When not using multipathing:

```yaml
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 99-worker-element-iscsi
spec:
  config:
    ignition:
      version: 3.2.0
    systemd:
      units:
      - name: iscsid.service
        enabled: true
        state: started
    osImageURL: ""
```

When using multipathing:
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: 99-worker-ontap-iscsi
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
        - contents:
            source: data:text/plain;charset=utf-8;base64,ZGVmYXVsdHMgewogICAgICAgIHVzZXJfZnJpZW5kbHlfbmFtZXMgYm9mbm8KICAICAgICAgIHByb3BlcnR5ICIoU0NTSV9JREVOVF98SURfV1dOKSIKfQoK
            verification:
        filesystem: root
        mode: 400
        path: /etc/multipath.conf
  systemd:
    units:
    - name: iscsid.service
      enabled: true
      state: started
    - name: multipathd.service
      enabled: true
      state: started
  osImageURL: ""

2. After the configuration is created, it takes approximately 20 to 30 minutes to apply the configuration to the worker nodes and reload them. Verify whether the machine config is applied by using `oc get mcp` and make sure that the machine config pool for workers is updated. You can also log into the worker nodes to confirm that the iscsid service is running (and the multipathd service is running if using multipathing).
It is also possible to confirm that the MachineConfig has been successfully applied and services have been started as expected by running the `oc debug` command with the appropriate flags.

Create storage-system backends

After completing the Astra Trident Operator install, you must configure the backend for the specific NetApp
To enable Trident integration with the NetApp ONTAP storage system, you must create a backend that enables communication with the storage system.

1. There are sample backend files available in the downloaded installation archive in the sample-input folder hierarchy. For NetApp ONTAP systems serving NFS, copy the backend-ontap-nas.json file to your working directory and edit the file.

   ```bash
   [netapp-user@rhel7 trident-installer]$ cp sample-input/backends-samples/ontap-nas/backend-ontap-nas.json ./
   [netapp-user@rhel7 trident-installer]$ vi backend-ontap-nas.json
   ```

2. Edit the backendName, managementLIF, dataLIF, svm, username, and password values in this file.

   ```json
   {
       "version": 1,
       "storageDriverName": "ontap-nas",
       "backendName": "ontap-nas+10.61.181.221",
       "managementLIF": "172.21.224.201",
       "dataLIF": "10.61.181.221",
       "svm": "trident_svm",
       "username": "cluster-admin",
       "password": "password"
   }
   ```

   It is a best practice to define the custom backendName value as a combination of the storageDriverName and the dataLIF that is serving NFS for easy identification.

3. With this backend file in place, run the following command to create your first backend.
4. With the backend created, you must next create a storage class. Just as with the backend, there is a sample storage class file that can be edited for the environment available in the sample-inputs folder. Copy it to the working directory and make necessary edits to reflect the backend created.

   ```bash
   [netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
   [netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
   ```

5. The only edit that must be made to this file is to define the `backendType` value to the name of the storage driver from the newly created backend. Also note the name-field value, which must be referenced in a later step.

   ```yaml
   apiVersion: storage.k8s.io/v1
   kind: StorageClass
   metadata:
     name: basic-csi
   provisioner: csi.trident.netapp.io
   parameters:
     backendType: "ontap-nas"
   ```

   There is an optional field called `fsType` that is defined in this file. This line can be deleted in NFS backends.

6. Run the `oc` command to create the storage class.

   ```bash
   [netapp-user@rhel7 trident-installer]$ oc create -f storage-class-basic.yaml
   storageclass.storage.k8s.io/basic-csi created
   ```
7. With the storage class created, you must then create the first persistent volume claim (PVC). There is a sample pvc-basic.yaml file that can be used to perform this action located in sample-inputs as well.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

8. The only edit that must be made to this file is ensuring that the storageClassName field matches the one just created. The PVC definition can be further customized as required by the workload to be provisioned.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
      storageClassName: basic-csi
```

9. Create the PVC by issuing the oc command. Creation can take some time depending on the size of the backing volume being created, so you can watch the process as it completes.

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created
```

```
[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME    STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
basic   Bound    pvc-b4370d37-0fa4-4c17-bd86-94f96c94b42d   1Gi
RWO            basic-csi      7s
```

**NetApp ONTAP iSCSI configuration**

To enable Trident integration with the NetApp ONTAP storage system, you must create a backend that enables communication with the storage system.

1. There are sample backend files available in the downloaded installation archive in the sample-input folder hierarchy. For NetApp ONTAP systems serving iSCSI, copy the backend-ontap-san.json file to your working directory and edit the file.
2. Edit the managementLIF, dataLIF, svm, username, and password values in this file.

```json
{
    "version": 1,
    "storageDriverName": "ontap-san",
    "managementLIF": "172.21.224.201",
    "dataLIF": "10.61.181.240",
    "svm": "trident_svm",
    "username": "admin",
    "password": "password"
}
```

3. With this backend file in place, run the following command to create your first backend.

```bash
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create backend -f backend-ontap-san.json
```

```
+------------------------+----------------+
+--------------------------------------+--------+---------+
|          NAME          | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
|------------------------+----------------|
| ontapsan_10.61.181.241 | ontap-san      | 6788533c-7fea-4a35-b797-fb9bb3322b91 | online |       0 |
|------------------------+----------------|
+--------------------------------------+--------+---------+
```

4. With the backend created, you must next create a storage class. Just as with the backend, there is a sample storage class file that can be edited for the environment available in the sample-inputs folder. Copy it to the working directory and make necessary edits to reflect the backend created.

```bash
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

5. The only edit that must be made to this file is to define the backendType value to the name of the storage driver from the newly created backend. Also note the name-field value, which must be referenced in a later step.
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"

There is an optional field called `fsType` that is defined in this file. In iSCSI backends, this value can be set to a specific Linux filesystem type (XFS, ext4, etc) or can be deleted to allow OpenShift to decide what filesystem to use.

6. Run the `oc` command to create the storage class.

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

7. With the storage class created, you must then create the first persistent volume claim (PVC). There is a sample `pvc-basic.yaml` file that can be used to perform this action located in sample-inputs as well.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

8. The only edit that must be made to this file is ensuring that the `storageClassName` field matches the one just created. The PVC definition can be further customized as required by the workload to be provisioned.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
  - ReadWriteOnce
resources:
  requests:
    storage: 1Gi
storageClassName: basic-csi
```

9. Create the PVC by issuing the `oc` command. Creation can take some time depending on the size of the backing volume being created, so you can watch the process as it completes.
To enable Trident integration with the NetApp Element storage system, you must create a backend that enables communication with the storage system using the iSCSI protocol.

1. There are sample backend files available in the downloaded installation archive in the sample-input folder hierarchy. For NetApp Element systems serving iSCSI, copy the `backend-solidfire.json` file to your working directory and edit the file.

   ```bash
   [netapp-user@rhel7 trident-installer]$ cp sample-input/backends-samples/solidfire/backend-solidfire.json ./
   [netapp-user@rhel7 trident-installer]$ vi ./backend-solidfire.json
   ```

   a. Edit the user, password, and MVIP value on the `EndPoint` line.

   b. Edit the `SVIP` value.

   ```json
   {
     "version": 1,
     "storageDriverName": "solidfire-san",
     "EndPoint": "https://trident:password@172.21.224.150/json-rpc/8.0",
     "SVIP": "10.61.180.200:3260",
     "TenantName": "trident",
     "Types": ["Type": "Bronze", "Qos": {"minIOPS": 1000, "maxIOPS": 2000, "burstIOPS": 4000}],
     {"Type": "Silver", "Qos": {"minIOPS": 4000, "maxIOPS": 6000, "burstIOPS": 8000}},
     {"Type": "Gold", "Qos": {"minIOPS": 6000, "maxIOPS": 8000, "burstIOPS": 10000}}
   }
   ```

2. With this back-end file in place, run the following command to create your first backend.
3. With the backend created, you must next create a storage class. Just as with the backend, there is a sample storage class file that can be edited for the environment available in the sample-inputs folder. Copy it to the working directory and make necessary edits to reflect the backend created.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

4. The only edit that must be made to this file is to define the `backendType` value to the name of the storage driver from the newly created backend. Also note the name-field value, which must be referenced in a later step.

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "solidfire-san"
```

There is an optional field called `fsType` that is defined in this file. In iSCSI backends, this value can be set to a specific Linux filesystem type (XFS, ext4, and so on), or it can be deleted to allow OpenShift to decide what filesystem to use.

5. Run the `oc` command to create the storage class.

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-basic.yaml
storageclass.storage.k8s.io/basic-csi created
```
6. With the storage class created, you must then create the first persistent volume claim (PVC). There is a sample `pvc-basic.yaml` file that can be used to perform this action located in sample-inputs as well.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

7. The only edit that must be made to this file is ensuring that the `storageClassName` field matches the one just created. The PVC definition can be further customized as required by the workload to be provisioned.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
    storage: 1Gi
  storageClassName: basic-csi
```

8. Create the PVC by issuing the `oc` command. Creation can take some time depending on the size of the backing volume being created, so you can watch the process as it completes.

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created
```

```
[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME    STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
basic   Bound    pvc-3445b5cc-df24-453d-a1e6-b484e874349d   1Gi
RWO            basic-csi      5s
```

**Advanced Configuration Options**

**Exploring load balancer options: Red Hat OpenShift with NetApp**

In most cases, Red Hat OpenShift makes applications available to the outside world through routes. A service is exposed by giving it an externally reachable hostname. The defined route and the endpoints identified by its service can be consumed by an OpenShift router to provide this named connectivity to external clients.

However in some cases, applications require the deployment and configuration of customized load balancers to expose the appropriate services. One example of this is NetApp Astra Control Center. To meet this need, we have evaluated a number of custom load balancer options. Their installation and configuration are described in
Installing MetalLB load balancers: Red Hat OpenShift with NetApp

This page lists the installation and configuration instructions for the MetalLB load balancer.

MetalLB is a self-hosted network load balancer installed on your OpenShift cluster that allows the creation of OpenShift services of type load balancer in clusters that do not run on a cloud provider. The two main features of MetalLB that work together to support LoadBalancer services are address allocation and external announcement.

MetalLB configuration options

Based on how MetalLB announces the IP address assigned to LoadBalancer services outside of the OpenShift cluster, it operates in two modes:

- **Layer 2 mode.** In this mode, one node in the OpenShift cluster takes ownership of the service and responds to ARP requests for that IP to make it reachable outside of the OpenShift cluster. Because only the node advertises the IP, it has a bandwidth bottleneck and slow failover limitations. For more information, see the documentation [here](#).

- **BGP mode.** In this mode, all nodes in the OpenShift cluster establish BGP peering sessions with a router and advertise the routes to forward traffic to the service IPs. The prerequisite for this is to integrate MetalLB with a router in that network. Owing to the hashing mechanism in BGP, it has certain limitation when IP-to-Node mapping for a service changes. For more information, refer to the documentation [here](#).

For the purpose of this document, we are configuring MetalLB in layer-2 mode.

Installing The MetalLB Load Balancer

1. Download the MetalLB resources.

   ```bash
   [netapp-user@rhel7 ~]$ wget https://raw.githubusercontent.com/metallb/metallb/v0.10.2/manifests/namespace.yaml
   [netapp-user@rhel7 ~]$ wget https://raw.githubusercontent.com/metallb/metallb/v0.10.2/manifests/metallb.yaml
   ```

2. Edit file `metallb.yaml` and remove `spec.template.spec.securityContext` from controller Deployment and the speaker DaemonSet.

   **Lines to be deleted:**
securityContext:
  runAsNonRoot: true
  runAsUser: 65534

3. Create the `metallb-system` namespace.

```
[netapp-user@rhel7 ~]$ oc create -f namespace.yaml
namespace/metallb-system created
```

4. Create the MetalLB CR.

```
[netapp-user@rhel7 ~]$ oc create -f metallb.yaml
podsecuritypolicy.policy/controller created
podsecuritypolicy.policy/speaker created
serviceaccount/controller created
serviceaccount/speaker created
clusterrole.rbac.authorization.k8s.io/metallb-system:controller created
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker created
role.rbac.authorization.k8s.io/config-watcher created
role.rbac.authorization.k8s.io/pod-lister created
role.rbac.authorization.k8s.io/controller created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker created
rolebinding.rbac.authorization.k8s.io/config-watcher created
rolebinding.rbac.authorization.k8s.io/pod-lister created
rolebinding.rbac.authorization.k8s.io/controller created
daemonset.apps/speaker created
deployment.apps/controller created
```

5. Before configuring the MetalLB speaker, grant the speaker DaemonSet elevated privileges so that it can perform the networking configuration required to make the load balancers work.

```
[netapp-user@rhel7 ~]$ oc adm policy add-scc-to-user privileged -n metallb-system -z speaker
clusterrole.rbac.authorization.k8s.io/system:openshift:scc:privileged
added: "speaker"
```

6. Configure MetalLB by creating a ConfigMap in the `metallb-system` namespace.
7. Now when loadbalancer services are created, MetalLB assigns an externalIP to the services and advertises the IP address by responding to ARP requests.

If you wish to configure MetalLB in BGP mode, skip step 6 above and follow the procedure in the MetalLB documentation here.

Installing F5 BIG-IP Load Balancers

F5 BIG-IP is an Application Delivery Controller (ADC) that offers a broad set of advanced production-grade traffic management and security services like L4-L7 load balancing, SSL/TLS offload, DNS, firewall and many more. These services drastically increase the availability, security and performance of your applications.

F5 BIG-IP can be deployed and consumed in various ways, on dedicated hardware, in the cloud, or as a virtual appliance on-premises. Refer to the documentation here to explore and deploy F5 BIG-IP as per requirement.

For efficient integration of F5 BIG-IP services with Red Hat OpenShift, F5 offers the BIG-IP Container Ingress Service (CIS). CIS is installed as a controller pod that watches OpenShift API for certain Custom Resource Definitions (CRDs) and manages the F5 BIG-IP system configuration. F5 BIG-IP CIS can be configured to control service types LoadBalancers and Routes in OpenShift.

Further, for automatic IP address allocation to service the type LoadBalancer, you can utilize the F5 IPAM controller. The F5 IPAM controller is installed as a controller pod that watches OpenShift API for LoadBalancer services with an ipamLabel annotation to allocate the IP address from a preconfigured pool.

This page lists the installation and configuration instructions for F5 BIG-IP CIS and IPAM controller. As a prerequisite, you must have an F5 BIG-IP system deployed and licensed. It must also be licensed for SDN services, which are included by default with the BIG-IP VE base license.

F5 BIG-IP can be deployed in standalone or cluster mode. For the purpose of this validation, F5 BIG-IP was deployed in standalone mode, but, for production purposes, it is preferred to have a cluster of BIG-IPs to avoid a single point of failure.
An F5 BIG-IP system can be deployed on dedicated hardware, in the cloud, or as a virtual appliance on-premises with versions greater than 12.x for it to be integrated with F5 CIS. For the purpose of this document, the F5 BIG-IP system was validated as a virtual appliance, for example using the BIG-IP VE edition.

Validated releases

<table>
<thead>
<tr>
<th>Technology</th>
<th>Software version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat OpenShift</td>
<td>4.6 EUS, 4.7</td>
</tr>
<tr>
<td>F5 BIG-IP VE edition</td>
<td>16.1.0</td>
</tr>
<tr>
<td>F5 Container Ingress Service</td>
<td>2.5.1</td>
</tr>
<tr>
<td>F5 IPAM Controller</td>
<td>0.1.4</td>
</tr>
<tr>
<td>F5 AS3</td>
<td>3.30.0</td>
</tr>
</tbody>
</table>

Installation

1. Install the F5 Application Services 3 extension to allow BIG-IP systems to accept configurations in JSON instead of imperative commands. Go to [F5 AS3 GitHub repository](https://github.com/f5networks/as3), and download the latest RPM file.

2. Log into F5 BIG-IP system, navigate to iApps > Package Management LX and click Import.

3. Click Choose File and select the downloaded AS3 RPM file, click OK, and then click Upload.

4. Confirm that the AS3 extension is installed successfully.

5. Next configure the resources required for communication between OpenShift and BIG-IP systems. First create a tunnel between OpenShift and the BIG-IP server by creating a VXLAN tunnel interface on the BIG-IP system for OpenShift SDN. Navigate to Network > Tunnels > Profiles, click Create, and set the Parent Profile to vxlan and the Flooding Type to Multicast. Enter a name for the profile and click Finished.
6. Navigate to Network > Tunnels > Tunnel List, click Create, and enter the name and local IP address for the tunnel. Select the tunnel profile that was created in the previous step and click Finished.

![Network tunnel creation screenshot](image)

7. Log into the Red Hat OpenShift cluster with cluster-admin privileges.

8. Create a hostsubnet on OpenShift for the F5 BIG-IP server, which extends the subnet from the OpenShift cluster to the F5 BIG-IP server. Download the host subnet YAML definition.

   ```bash
   wget https://github.com/F5Networks/k8s-bigip-ctlr/blob/master/docs/config_examples/openshift/f5-kctlr-openshift-hostsubnet.yaml
   ```

9. Edit the host subnet file and add the BIG-IP VTEP (VXLAN tunnel) IP for the OpenShift SDN.
apiVersion: v1
class: HostSubnet
metadata:
  name: f5-server
  annotations:
    pod.network.openshift.io/fixed-vnid-host: "0"
    pod.network.openshift.io/assign-subnet: "true"
# provide a name for the node that will serve as BIG-IP's entry into the
# cluster
host: f5-server
# The hostIP address will be the BIG-IP interface address routable to
# OpenShift Origin nodes.
# This address is the BIG-IP VTEP in the SDN's VXLAN.
hostIP: 10.63.172.239

- Change the hostIP and other details as applicable to your environment.

10. Create the HostSubnet resource.

    [admin@rhel-7 ~]$ oc create -f f5-kctlr-openshift-hostsubnet.yaml

    subnet.network.openshift.io/f5-server created

11. Get the cluster IP subnet range for the host subnet created for the F5 BIG-IP server.
<table>
<thead>
<tr>
<th>NAME</th>
<th>HOST</th>
<th>HOST IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.131.0.0/23</td>
<td>f5-server</td>
<td>10.63.172.239</td>
</tr>
<tr>
<td>ocp-vmw-nszws-master-0</td>
<td>ocp-vmw-nszws-master-0</td>
<td>10.63.172.44</td>
</tr>
<tr>
<td>10.128.0.0/23</td>
<td>ocp-vmw-nszws-master-1</td>
<td>10.63.172.47</td>
</tr>
<tr>
<td>ocp-vmw-nszws-master-2</td>
<td>ocp-vmw-nszws-master-2</td>
<td>10.63.172.48</td>
</tr>
<tr>
<td>10.129.0.0/23</td>
<td>ocp-vmw-nszws-worker-r8fh4</td>
<td>10.63.172.7</td>
</tr>
<tr>
<td>10.130.2.0/23</td>
<td>ocp-vmw-nszws-worker-tvr46</td>
<td>10.63.172.11</td>
</tr>
<tr>
<td>10.128.2.0/23</td>
<td>ocp-vmw-nszws-worker-w8r4</td>
<td>10.63.172.15</td>
</tr>
<tr>
<td>10.131.2.0/23</td>
<td>ocp-vmw-nszws-worker-wtgfw</td>
<td>10.63.172.17</td>
</tr>
</tbody>
</table>

12. Create a self IP on OpenShift VXLAN with an IP in OpenShift's host subnet range corresponding to the F5 BIG-IP server. Log into the F5 BIG-IP system, navigate to Network > Self IPs and click Create. Enter an IP from the cluster IP subnet created for F5 BIG-IP host subnet, select the VXLAN tunnel, and enter the other details. Then click Finished.
13. Create a partition in the F5 BIG-IP system to be configured and used with CIS. Navigate to System > Users > Partition List, click Create, and enter the details. Then click Finished.

![](image)

---

F5 recommends that no manual configuration be done on the partition that is managed by CIS.

14. Install the F5 BIG-IP CIS using the operator from OperatorHub. Log into the Red Hat OpenShift cluster with cluster-admin privileges and create a secret with F5 BIG-IP system login credentials, which is a prerequisite for the operator.
15. Install the F5 CIS CRDs.

```bash
[admin@rhel-7 ~]$ oc create secret generic bigip-login -n kube-system --from-literal=username=admin --from-literal=password=admin
secret/bigip-login created
```

```bash
[admin@rhel-7 ~]$ oc apply -f https://raw.githubusercontent.com/F5Networks/k8s-bigip-ctlr/master/docs/config_examples/crd/Install/customresourcedefinitions.yaml

customresourcedefinition.apiextensions.k8s.io/virtualservers.cis.f5.com created

customresourcedefinition.apiextensions.k8s.io/tlsprofiles.cis.f5.com created

customresourcedefinition.apiextensions.k8s.io/transportservers.cis.f5.com created

customresourcedefinition.apiextensions.k8s.io/externaldnss.cis.f5.com created

customresourcedefinition.apiextensions.k8s.io/ingresslinks.cis.f5.com created
```

16. Navigate to Operators > OperatorHub, search for the keyword F5, and click the F5 Container Ingress Service tile.

**OperatorHub**

Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. You can purchase commercial software through [Red Hat Marketplace](https://marketplace.redhat.com). You can install Operators on your clusters to provide optional add-ons and shared services to your developers. After installation, the Operator capabilities will appear in the Developer Catalog providing a self-service experience.

![OperatorHub](image)
17. Read the operator information and click Install.

18. On the Install operator screen, leave all default parameters, and click Install.
19. It takes a while to install the operator.

20. After the operator is installed, the Installation Successful message is displayed.

21. Navigate to Operators > Installed Operators, click F5 Container Ingress Service, and then click Create Instance under the F5BigIpCtlr tile.
22. Click YAML View and paste the following content after updating the necessary parameters.

```
Update the parameters bigip_partition, 'openshift_sdn_name', bigip_url and bigip_login_secret below to reflect the values for your setup before copying the content.
```
apiVersion: cis.f5.com/v1
kind: F5BigIpCtlr
metadata:
  name: f5-server
  namespace: openshift-operators
spec:
  args:
    log_as3_response: true
    agent: as3
    log_level: DEBUG
    bigip_partition: ocp-vmw
    openshift_sdn_name: /Common/openshift_vxlan
    bigip_url: 10.61.181.19
    insecure: true
    pool-member-type: cluster
    custom_resource_mode: true
    as3_validation: true
    ipam: true
    manage_configmaps: true
  bigip_login_secret: bigip-login
  image:
    pullPolicy: Always
    repo: f5networks/cntr-ingress-svcs
    user: registry.connect.redhat.com
  namespace: kube-system
  rbac:
    create: true
    resources: {}
  serviceAccount:
    create: true
  version: latest

23. After pasting this content, click Create. This installs the CIS pods in the kube-system namespace.
Red Hat OpenShift, by default, provides a way to expose the services via Routes for L7 load balancing. An inbuilt OpenShift router is responsible for advertising and handling traffic for these routes. However, you can also configure the F5 CIS to support the Routes through an external F5 BIG-IP system, which can run either as an auxiliary router or a replacement to the self-hosted OpenShift router. CIS creates a virtual server in the BIG-IP system that acts as a router for the OpenShift routes, and BIG-IP handles the advertisement and traffic routing. Refer to the documentation here for information on parameters to enable this feature. Note that these parameters are defined for OpenShift Deployment resource in the apps/v1 API. Therefore, when using these with the F5BigIpCtbl resource cis.f5.com/v1 API, replace the hyphens (-) with underscores (_) for the parameter names.

24. The arguments that are passed to the creation of CIS resources include `ipam: true` and `custom_resource_mode: true`. These parameters are required for enabling CIS integration with an IPAM controller. Verify that the CIS has enabled IPAM integration by creating the F5 IPAM resource.

```
[admin@rhel-7 ~]$ oc get f5ipam -n kube-system
```

<table>
<thead>
<tr>
<th>NAMESPACE</th>
<th>NAME</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>kube-system</td>
<td>ipam.10.61.181.19.ocp-vmw</td>
<td>43s</td>
</tr>
</tbody>
</table>

25. Create the service account, role and rolebinding required for the F5 IPAM controller. Create a YAML file and paste the following content.
26. Create the resources.

```bash
[admin@rhel-7 ~]$ vi f5-ipam-rbac.yaml

kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: ipam-ctlr-clusterrole
rules:
- apiGroups: ["fic.f5.com"]
  resources: ["ipams","ipams/status"]
  verbs: ["get", "list", "watch", "update", "patch"]
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: ipam-ctlr-clusterrole-binding
  namespace: kube-system
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: ipam-ctlr-clusterrole
subjects:
- apiGroup: ""
  kind: ServiceAccount
  name: ipam-ctlr
  namespace: kube-system
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: ipam-ctlr
  namespace: kube-system

[admin@rhel-7 ~]$ oc create -f f5-ipam-rbac.yaml

clusterrole.rbac.authorization.k8s.io/ipam-ctlr-clusterrole created
clusterrolebinding.rbac.authorization.k8s.io/ipam-ctlr-clusterrole-binding created
serviceaccount/ipam-ctlr created
```

27. Create a YAML file and paste the F5 IPAM deployment definition provided below.
Update the ip-range parameter in spec.template.spec.containers[0].args below to reflect the ipamLabels and IP address ranges corresponding to your setup.

ipamLabels [range1 and range2 in below example] are required to be annotated for the services of type LoadBalancer for the IPAM controller to detect and assign an IP address from the defined range.

```
$ vi f5-ipam-deployment.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
    labels:
        name: f5-ipam-controller
        name: f5-ipam-controller
        namespace: kube-system
spec:
    replicas: 1
    selector:
        matchLabels:
            app: f5-ipam-controller
    template:
        metadata:
            creationTimestamp: null
            labels:
                app: f5-ipam-controller
        spec:
            containers:
            - args:
              --orchestration=openshift
              --ip-range='{"range1":"10.63.172.242-10.63.172.249",
              "range2":"10.63.170.111-10.63.170.129"}'
              --log-level=DEBUG
            command:
            - /app/bin/f5-ipam-controller
            image: registry.connect.redhat.com/f5networks/f5-ipam-controller:latest
            imagePullPolicy: IfNotPresent
            name: f5-ipam-controller
            dnsPolicy: ClusterFirst
            restartPolicy: Always
            schedulerName: default-scheduler
            securityContext: {}
            serviceAccount: ipam-ctlr
            serviceAccountName: ipam-ctlr
```
28. Create the F5 IPAM controller deployment.

   [admin@rhel-7 ~]$ oc create -f f5-ipam-deployment.yaml
   deployment/f5-ipam-controller created

29. Verify the F5 IPAM controller pods are running.

   [admin@rhel-7 ~]$ oc get pods -n kube-system
   
   NAME                                       READY   STATUS    RESTARTS AGE
   f5-ipam-controller-5986cff5bd-2bvn6        1/1     Running   0          30s
   f5-server-f5-bigip-ctlr-5d7578667d-qxdgj   1/1     Running   0          14m

30. Create the F5 IPAM schema.

   [admin@rhel-7 ~]$ oc create -f
   https://raw.githubusercontent.com/F5Networks/f5-ipam-controller/main/docs/_static/schemas/ipam_schema.yaml
   customresourcedefinition.apiextensions.k8s.io/ipams.fic.f5.com

Verification

1. Create a service of type LoadBalancer
2. Check if the IPAM controller assigns an external IP to it.

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLUSTER-IP</th>
<th>EXTERNAL-IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>80:32605/TCP</td>
<td>AGE</td>
<td>27s</td>
<td></td>
</tr>
</tbody>
</table>

3. Create a deployment and use the LoadBalancer service that was created.
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: f5-demo-test
    name: f5-demo-test
spec:
  replicas: 2
  selector:
    matchLabels:
      app: f5-demo-test
  template:
    metadata:
      labels:
        app: f5-demo-test
    spec:
      containers:
      - env:
        - name: service_name
          value: f5-demo-test
        image: nginx
        imagePullPolicy: Always
        name: f5-demo-test
        ports:
        - containerPort: 80
          protocol: TCP
```

```
[admin@rhel-7 ~]$ vi example_deployment.yaml

[admin@rhel-7 ~]$ oc create -f example_deployment.yaml

deployment/f5-demo-test created

4. Check if the pods are running.

   [admin@rhel-7 ~]$ oc get pods

   NAME                        READY STATUS    RESTARTS AGE
   f5-demo-test-57c46f6f98-47wwp 1/1   Running   0   27s
   f5-demo-test-57c46f6f98-cl2m8 1/1   Running   0   27s

5. Check if the corresponding virtual server is created in the BIG-IP system for the service of type LoadBalancer in OpenShift. Navigate to Local Traffic > Virtual Servers > Virtual Server List.
```
Creating Private Image Registries

For most deployments of Red Hat OpenShift, using a public registry like Quay.io or DockerHub meets most customer’s needs. However there are times when a customer may want to host their own private or customized images.

This procedure documents creating a private image registry which is backed by a persistent volume provided by Astra Trident and NetApp ONTAP.

Astra Control Center requires a registry to host the images the Astra containers require. The following section describes the steps to setup a private registry on Red Hat OpenShift cluster and pushing the images required to support the installation of Astra Control Center.

Creating A private image registry

1. Remove the default annotation from the current default storage class and annotate the Trident-backed storage class as default for the OpenShift cluster.

   ```bash
   [netapp-user@rhel7 ~]$ oc patch storageclass thin -p '{"metadata":
   {"annotations": {"storageclass.kubernetes.io/is-default-class":
   "false"}}}'
   storageclass.storage.k8s.io/thin patched
   
   [netapp-user@rhel7 ~]$ oc patch storageclass ocp-trident -p
   '{"metadata": {"annotations": {"storageclass.kubernetes.io/is-default-class": "true"}}}'
   storageclass.storage.k8s.io/ocp-trident patched
   ```

2. Edit the imageregistry operator by entering the following storage parameters in the `spec` section.

   ```bash
   [netapp-user@rhel7 ~]$ oc edit
   configs.imageregistry.operator.openshift.io

   storage:
   pvc:
   claim:
   ```

3. Enter the following parameters in the `spec` section for creating a OpenShift route with a custom hostname. Save and exit.
The above route config is used when you want a custom hostname for your route. If you want OpenShift to create a route with a default hostname, you can add the following parameters to the `spec` section: `defaultRoute: true`.

### Custom TLS certificates

When you are using a custom hostname for the route, by default, it uses the default TLS configuration of the OpenShift Ingress operator. However, you can add a custom TLS configuration to the route. To do so, complete the following steps.

#### a. Create a secret with the route’s TLS certificates and key.

```
[netapp-user@rhel7 ~]$ oc create secret tls astra-route-tls -n openshift-image-registry –cert/home/admin/netapp-astra/tls.crt --key=/home/admin/netapp-astra/tls.key
```

#### b. Edit the `imageregistry` operator and add the following parameters to the `spec` section.

```
[netapp-user@rhel7 ~]$ oc edit
configs.imageregistry.operator.openshift.io

routes:
  - hostname: astra-registry.apps.ocp-vmw.cie.netapp.com
    name: netapp-astra-route
    secretName: astra-route-tls
```

4. Edit the `imageregistry` operator again and change the management state of the operator to the `Managed` state. Save and exit.

```
oc edit configs.imageregistry/cluster

managementState: Managed
```

5. If all the prerequisites are satisfied, PVCs, pods, and services are created for the private image registry. In a few minutes, the registry should be up.

```
[netapp-user@rhel7 ~]$oc get all -n openshift-image-registry
```
<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>pod/cluster-image-registry-operator-74f6d954b6-rb7zr</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td>pod/image-pruner-1627257600-f5cpj</td>
<td>0/1</td>
<td>Completed</td>
</tr>
<tr>
<td>pod/image-pruner-1627344000-swqx9</td>
<td>0/1</td>
<td>Completed</td>
</tr>
<tr>
<td>pod/image-pruner-1627430400-rv5nt</td>
<td>0/1</td>
<td>Completed</td>
</tr>
<tr>
<td>pod/image-registry-6758b547f-6pnj8</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td>pod/node-ca-bwb5r</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td>pod/node-ca-f8w54</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td>pod/node-ca-gjx7h</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td>pod/node-ca-lcx4k</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td>pod/node-ca-v7zmx</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td>pod/node-ca-xpppp</td>
<td>1/1</td>
<td>Running</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLUSTER-IP</th>
<th>EXTERNAL-IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>service/image-registry</td>
<td>ClusterIP</td>
<td>172.30.196.167</td>
<td>&lt;none&gt;</td>
</tr>
<tr>
<td>service/image-registry-operator</td>
<td>ClusterIP</td>
<td>None</td>
<td>&lt;none&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESIRED</th>
<th>CURRENT</th>
<th>READY</th>
<th>UP-TO-DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>daemonset.apps/node-ca</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>kubernetes.io/os=linux</td>
<td>90d</td>
<td>90d</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>UP-TO-DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>deployment.apps/cluster-image-registry-operator</td>
<td>1/1</td>
<td>1</td>
</tr>
<tr>
<td>deployment.apps/image-registry</td>
<td>1/1</td>
<td>1</td>
</tr>
</tbody>
</table>
6. If you are using the default TLS certificates for the ingress operator OpenShift registry route, you can fetch the TLS certificates using the following command.

```
[netapp-user@rhel7 ~]$ oc extract secret/router-ca --keys=tls.crt -n openshift-ingress-operator
```

7. To allow OpenShift nodes to access and pull the images from the registry, add the certificates to the docker client on the OpenShift nodes. Create a configmap in the openshift-config namespace using the TLS certificates and patch it to the cluster image config to make the certificate trusted.
8. The OpenShift internal registry is controlled by authentication. All the OpenShift users can access the OpenShift registry, but the operations that the logged in user can perform depends on the user permissions.

   a. To allow a user or a group of users to pull images from the registry, the user(s) must have the registry-viewer role assigned.

   ```bash
   [netapp-user@rhel7 ~]$ oc policy add-role-to-user registry-viewer ocp-user
   [netapp-user@rhel7 ~]$ oc policy add-role-to-group registry-viewer ocp-user-group
   ``

   b. To allow a user or group of users to write or push images, the user(s) must have the registry-editor role assigned.

   ```bash
   [netapp-user@rhel7 ~]$ oc policy add-role-to-user registry-editor ocp-user
   [netapp-user@rhel7 ~]$ oc policy add-role-to-group registry-editor ocp-user-group
   ```

9. For OpenShift nodes to access the registry and push or pull the images, you need to configure a pull secret.

   ```bash
   [netapp-user@rhel7 ~]$ oc create secret docker-registry astra-registry-credentials --docker-server=astra-registry.apps.ocp-vmw.cie.netapp.com --docker-username=ocp-user --docker-password=password
   ```

10. This pull secret can then be patched to serviceaccounts or be referenced in the corresponding pod definition.

    a. To patch it to service accounts, run the following command.

    ```bash
    [netapp-user@rhel7 ~]$ oc secrets link <service_account_name> astra-registry-credentials --for=pull
    ```
b. To reference the pull secret in the pod definition, add the following parameter to the `spec` section.

```yaml
imagePullSecrets:
  - name: astra-registry-credentials
```

11. To push or pull an image from workstations apart from OpenShift node, complete the following steps.

a. Add the TLS certificates to the docker client.

```bash
[netapp-user@rhel7 ~]$ sudo mkdir /etc/docker/certs.d/astra-registry.apps.ocp-vmw.cie.netapp.com

[netapp-user@rhel7 ~]$ sudo cp /path/to/tls.crt /etc/docker/certs.d/astra-registry.apps.ocp-vmw.cie.netapp.com
```

b. Log into OpenShift using the `oc login` command.

```bash
[netapp-user@rhel7 ~]$ oc login --token=sha256~D49SpB_lEsSrJYwrM0LIO-VRcjWHu0a27vKa0 --server=https://api.ocp-vmw.cie.netapp.com:6443
```

c. Log into the registry using OpenShift user credentials with the `podman/docker` command.

```
# podman
[netapp-user@rhel7 ~]$ podman login astra-registry.apps.ocp-vmw.cie.netapp.com -u kubemadmin -p $(oc whoami -t) --tls --verify=false

# docker
[netapp-user@rhel7 ~]$ docker login astra-registry.apps.ocp-vmw.cie.netapp.com -u kubemadmin -p $(oc whoami -t)
```

+ **NOTE:** If you are using `kubeadmin` user to log into the private registry, then use token instead of password.

```
+ **NOTE:** If you are using `kubeadmin` user to log into the private registry, then use token instead of password.
```

d. Push or pull the images.
Solution Validation and Use Cases: Red Hat OpenShift with NetApp

The examples provided on this page are solution validations and use cases for Red Hat OpenShift with NetApp.

- Deploy a Jenkins CI/CD Pipeline with Persistent Storage
- Configure Multitenancy on Red Hat OpenShift with NetApp
- Red Hat OpenShift Virtualization with NetApp ONTAP
- Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp

Deploy a Jenkins CI/CD Pipeline with Persistent Storage: Red Hat OpenShift with NetApp

This section provides the steps to deploy a continuous integration/continuous delivery or deployment (CI/CD) pipeline with Jenkins to validate solution operation.

Create the resources required for Jenkins deployment

To create the resources required for deploying the Jenkins application, complete the following steps:

1. Create a new project named Jenkins.
2. In this example, we deployed Jenkins with persistent storage. To support the Jenkins build, create the PVC. Navigate to Storage > Persistent Volume Claims and click Create Persistent Volume Claim. Select the storage class that was created, make sure that the Persistent Volume Claim Name is jenkins, select the appropriate size and access mode, and then click Create.
To deploy Jenkins with persistent storage, complete the following steps:

1. In the upper left corner, change the role from Administrator to Developer. Click +Add and select From Catalog. In the Filter by Keyword bar, search for jenkins. Select Jenkins Service with Persistent Storage.
2. Click **Instantiate Template**.

![Jenkins template](jenkins_template.png)

**Jenkins**

Provided by Red Hat, Inc.

**Instantiate Template**

<table>
<thead>
<tr>
<th>Provider</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat, Inc.</td>
<td>Jenkins service, with persistent storage. NOTE: You must have persistent volumes available in your cluster to use this template.</td>
</tr>
</tbody>
</table>

**Support**

Get support

**Documentation**

[https://docs.okd.io/latest/using_images/other_images/jenkins.html](https://docs.okd.io/latest/using_images/other_images/jenkins.html)

3. By default, the details for the Jenkins application are populated. Based on your requirements, modify the parameters and click Create. This process creates all the required resources for supporting Jenkins on
OpenShift.

4. The Jenkins pods take approximately 10 to 12 minutes to enter the Ready state.
5. After the pods are instantiated, navigate to Networking > Routes. To open the Jenkins webpage, click the URL provided for the jenkins route.

6. Because OpenShift OAuth was used while creating the Jenkins app, click Log in with OpenShift.
7. Authorize the Jenkins service account to access the OpenShift users.

Authorize Access

Service account jenkins in project jenkins is requesting permission to access your account (kube:admin)

Requested permissions

- **user:info**
  - Read-only access to your user information (including username, identities, and group membership)

- **user:check-access**
  - Read-only access to view your privileges (for example, “can I create builds?”)

You will be redirected to https://jenkins-jenkins.apps.rhv-ocp-cluster.cle.netapp.com/securityRealm/finishLogin

[Allow selected permissions]  [Deny]

8. The Jenkins welcome page is displayed. Because we are using a Maven build, complete the Maven installation first. Navigate to Manage Jenkins > Global Tool Configuration, and then, in the Maven subhead, click Add Maven. Enter the name of your choice and make sure that the Install Automatically option is selected. Click Save.
9. You can now create a pipeline to demonstrate the CI/CD workflow. On the home page, click Create New Jobs or New Item from the left-hand menu.

10. On the Create Item page, enter the name of your choice, select Pipeline, and click Ok.

11. Select the Pipeline tab. From the Try Sample Pipeline drop-down menu, select Github + Maven. The code is automatically populated. Click Save.
12. Click Build Now to trigger the development through the preparation, build, and testing phase. It can take several minutes to complete the whole build process and display the results of the build.
13. Whenever there are any code changes, the pipeline can be rebuilt to patch the new version of software enabling continuous integration and continuous delivery. Click Recent Changes to track the changes from the previous version.
Configure Multi-tenancy on Red Hat OpenShift with NetApp ONTAP

Configuring multitenancy on Red Hat OpenShift with NetApp

Many organizations that run multiple applications or workloads on containers tend to deploy one Red Hat OpenShift cluster per application or workload. This allows them to implement strict isolation for the application or workload, optimize performance, and reduce security vulnerabilities. However, deploying a separate Red Hat OpenShift cluster for each application poses its own set of problems. It increases operational overhead having to monitor and manage each cluster on its own, increases cost owing to dedicated resources for different applications, and hinders efficient scalability.

To overcome these problems, one can consider running all the applications or workloads in a single Red Hat OpenShift cluster. But in such an architecture, resource isolation and application security vulnerabilities are one of the major challenges. Any security vulnerability in one workload could naturally spill over into another workload, thus increasing the impact zone. In addition, any abrupt uncontrolled resource utilization by one application can affect the performance of another application, because there is no resource allocation policy by default.

Therefore, organizations look out for solutions that pick up the best in both worlds, for example, by allowing them to run all their workloads in a single cluster and yet offering the benefits of a dedicated cluster for each workload.

One such effective solution is to configure multitenancy on Red Hat OpenShift. Multitenancy is an architecture
that allows multiple tenants to coexist on the same cluster with proper isolation of resources, security, and so on. In this context, a tenant can be viewed as a subset of the cluster resources that are configured to be used by a particular group of users for an exclusive purpose. Configuring multitenancy on a Red Hat OpenShift cluster provides the following advantages:

- A reduction in CapEx and OpEx by allowing cluster resources to be shared
- Lower operational and management overhead
- Securing the workloads from cross-contamination of security breaches
- Protection of workloads from unexpected performance degradation due to resource contention

For a fully realized multitenant OpenShift cluster, quotas and restrictions must be configured for cluster resources belonging to different resource buckets: compute, storage, networking, security, and so on. Although we cover certain aspects of all the resource buckets in this solution, we focus on best practices for isolating and securing the data served or consumed by multiple workloads on the same Red Hat OpenShift cluster by configuring multitenancy on storage resources that are dynamically allocated by Astra Trident backed by NetApp ONTAP.

Architecture

Although Red Hat OpenShift and Astra Trident backed by NetApp ONTAP do not provide isolation between workloads by default, they offer a wide range of features that can be used to configure multitenancy. To better understand designing a multitenant solution on a Red Hat OpenShift cluster with Astra Trident backed by NetApp ONTAP, let us consider an example with a set of requirements and outline the configuration around it.

Let us assume that an organization runs two of its workloads on a Red Hat OpenShift cluster as part of two projects that two different teams are working on. The data for these workloads reside on PVCs that are dynamically provisioned by Astra Trident on a NetApp ONTAP NAS backend. The organization has a requirement to design a multitenant solution for these two workloads and isolate the resources used for these projects to make sure that security and performance is maintained, primarily focused on the data that serves those applications.

The following figure depicts the multitenant solution on a Red Hat OpenShift cluster with Astra Trident backed by NetApp ONTAP.
Technology requirements

1. NetApp ONTAP storage cluster
2. Red Hat OpenShift cluster
3. Astra Trident

Red Hat OpenShift – Cluster resources

From the Red Hat OpenShift cluster point of view, the top-level resource to start with is the project. An OpenShift project can be viewed as a cluster resource that divides the whole OpenShift cluster into multiple virtual clusters. Therefore, isolation at project level provides a base for configuring multitenancy.

Next up is to configure RBAC in the cluster. The best practice is to have all the developers working on a single project or workload configured into a single user group in the Identity Provider (IdP). Red Hat OpenShift allows IdP integration and user group synchronization thus allowing the users and groups from the IdP to be imported into the cluster. This helps the cluster administrators to segregate access of the cluster resources dedicated to a project to a user group or groups working on that project, thereby restricting unauthorized access to any cluster resources. To learn more about IdP integration with Red Hat OpenShift, see the documentation here.

NetApp ONTAP

It is important to isolate the shared storage serving as a persistent storage provider for a Red Hat OpenShift cluster to make sure that the volumes created on the storage for each project appear to the hosts as if they are
created on separate storage. To do this, create as many SVMs (storage virtual machines) on NetApp ONTAP as there are projects or workloads, and dedicate each SVM to a workload.

Astra Trident

After you have different SVMs for different projects created on NetApp ONTAP, you must map each SVM to a different Trident backend. The backend configuration on Trident drives the allocation of persistent storage to OpenShift cluster resources, and it requires the details of the SVM to be mapped to. This should be the protocol driver for the backend at the minimum. Optionally, it allows you to define how the volumes are provisioned on the storage and to set limits for the size of volumes or usage of aggregates and so on. Details concerning the definition of the Trident backends can be found [here](#).

Red Hat OpenShift – storage resources

After configuring the Trident backends, the next step is to configure StorageClasses. Configure as many storage classes as there are backends, providing each storage class access to spin up volumes only on one backend. We can map the StorageClass to a particular Trident backend by using the storagePools parameter while defining the storage class. The details to define a storage class can be found [here](#). Thus, there is a one-to-one mapping from StorageClass to Trident backend which points back to one SVM. This ensures that all storage claims via the StorageClass assigned to that project are served by the SVM dedicated to that project only.

Because storage classes are not namespaced resources, how do we ensure that storage claims to storage class of one project by pods in another namespace or project gets rejected? The answer is to use ResourceQuotas. ResourceQuotas are objects that control the total usage of resources per project. It can limit the number as well as the total amount of resources that can be consumed by objects in the project. Almost all the resources of a project can be limited using ResourceQuotas and using this efficiently can help organizations cut cost and outages due to overprovisioning or overconsumption of resources. Refer to the documentation [here](#) for more information.

For this use case, we need to limit the pods in a particular project from claiming storage from storage classes that are not dedicated to their project. To do that, we need to limit the persistent volume claims for other storage classes by setting `<storage-class-name>.storageclass.storage.k8s.io/persistentvolumeclaims` to 0. In addition, a cluster administrator must ensure that the developers in a project should not have access to modify the ResourceQuotas.

Configuration

For any multitenant solution, no user can have access to more cluster resources than is required. So, the entire set of resources that are to be configured as part of the multitenancy configuration is divided between cluster-admin, storage-admin, and developers working on each project.

The following table outlines the different tasks to be performed by different users:
<table>
<thead>
<tr>
<th>Role</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster-admin</td>
<td>Create projects for different applications or workloads</td>
</tr>
<tr>
<td></td>
<td>Create ClusterRoles and RoleBindings for storage-admin</td>
</tr>
<tr>
<td></td>
<td>Create Roles and RoleBindings for developers assigning access to specific projects</td>
</tr>
<tr>
<td></td>
<td>[Optional] Configure projects to schedule pods on specific nodes</td>
</tr>
<tr>
<td>Storage-admin</td>
<td>Create SVMs on NetApp ONTAP</td>
</tr>
<tr>
<td></td>
<td>Create Trident backends</td>
</tr>
<tr>
<td></td>
<td>Create StorageClasses</td>
</tr>
<tr>
<td></td>
<td>Create storage ResourceQuotas</td>
</tr>
<tr>
<td>Developers</td>
<td>Validate access to create or patch PVCs or pods in assigned project</td>
</tr>
<tr>
<td></td>
<td>Validate access to create or patch PVCs or pods in another project</td>
</tr>
<tr>
<td></td>
<td>Validate access to view or edit Projects, ResourceQuotas, and StorageClasses</td>
</tr>
</tbody>
</table>

**Configuration**

**Prerequisites**

- NetApp ONTAP cluster
- Red Hat OpenShift cluster
- Trident installed on the cluster
- Admin workstation with tridentctl and oc tools installed and added to $PATH
- Admin access to ONTAP
- Cluster-admin access to OpenShift cluster
- Cluster is integrated with Identity Provider
- Identity provider is configured to efficiently distinguish between users in different teams

**Configuration: cluster-admin tasks**

The following tasks are performed by the Red Hat OpenShift cluster-admin:

1. Log into Red Hat OpenShift cluster as the cluster-admin.
2. Create two projects corresponding to different projects.

```bash
oc create namespace project-1
oc create namespace project-2
```
3. Create the developer role for project-1.

```bash
cat << EOF | oc create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: project-1
  name: developer-project-1
rules:
- verbs:
  - '*'
    apiGroups:
    - apps
    - batch
    - autoscaling
    - extensions
    - networking.k8s.io
    - policy
    - apps.openshift.io
    - build.openshift.io
    - image.openshift.io
    - ingress.operator.openshift.io
    - route.openshift.io
    - snapshot.storage.k8s.io
    - template.openshift.io
  resources:
  - '*'
- verbs:
  - '*'
    apiGroups:
    - '
    resources:
    - bindings
    - configmaps
    - endpoints
    - events
    - persistentvolumelclaims
    - pods
    - pods/log
    - pods/attach
    - podtemplates
    - replicationcontrollers
    - services
    - limitranges
    - namespaces
    - componentstatuses
```
The role definition provided in this section is just an example. Developer roles must be defined based on end-user requirements.

4. Similarly, create developer roles for project-2.

5. All OpenShift and NetApp storage resources are usually managed by a storage admin. Access for storage administrators is controlled by the trident operator role that is created when Trident is installed. In addition to this, the storage admin also requires access to ResourceQuotas to control how storage is consumed.

6. Create a role for managing ResourceQuotas in all projects in the cluster to attach it to storage admin.

```bash
cat << EOF | oc create -f -
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: resource-quotas-role
rules:
- verbs:
  - '*'
    apiGroups:
    - ''
    resources:
    - resourcequotas
- verbs:
  - '*'
    apiGroups:
    - quota.openshift.io
    resources:
    - '*'
EOF```

7. Make sure that the cluster is integrated with the organization’s identity provider and that user groups are synchronized with cluster groups. The following example shows that the identity provider has been integrated with the cluster and synchronized with the user groups.

```
# cat << EOF | oc create -f -
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: netapp-storage-admin-trident-operator
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-netapp-storage-admins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: trident-operator
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: netapp-storage-admin-resource-quotas-cr
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-netapp-storage-admins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: resource-quotas-role
EOF
```

For storage admins, two roles must be bound: trident-operator and resource-quotas.

9. Create RoleBindings for developers binding the developer-project-1 role to the corresponding group (ocp-project-1) in project-1.
cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-1-developer
  namespace: project-1
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-project-1
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: developer-project-1
EOF

10. Similarly, create RoleBindings for developers binding the developer roles to the corresponding user group in project-2.

**Configuration: Storage-admin tasks**

The following resources must be configured by a storage administrator:

1. Log into the NetApp ONTAP cluster as admin.

2. Navigate to Storage > Storage VMs and click Add. Create two SVMs, one for project-1 and the other for project-2, by providing the required details. Also create a vsadmin account to manage the SVM and its resources.
3. Log into the Red Hat OpenShift cluster as the storage administrator.

4. Create the backend for project-1 and map it to the SVM dedicated to the project. NetApp recommends using the SVM’s vsadmin account to connect the backend to SVM instead of using the ONTAP cluster administrator.
We are using the ontap-nas driver for this example. Use the appropriate driver when creating the backend based on the use case.

We assume that Trident is installed in the trident project.

5. Similarly create the Trident backend for project-2 and map it to the SVM dedicated to project-2.

6. Next, create the storage classes. Create the storage class for project-1 and configure it to use the storage pools from backend dedicated to project-1 by setting the storagePools parameter.

7. Likewise, create a storage class for project-2 and configure it to use the storage pools from backend dedicated to project-2.

8. Create a ResourceQuota to restrict resources in project-1 requesting storage from storageclasses dedicated to other projects.
9. Similarly, create a ResourceQuota to restrict resources in project-2 requesting storage from storageclasses dedicated to other projects.

**Validation**

To validate the multitenant architecture that was configured in the previous steps, complete the following steps:

**Validate access to create PVCs or pods in assigned project**

1. Log in as ocp-project-1-user, developer in project-1.
2. Check access to create a new project.

   ```
   oc create ns sub-project-1
   ```

3. Create a PVC in project-1 using the storageclass that is assigned to project-1.

   ```
   cat << EOF | oc create -f -
   kind: PersistentVolumeClaim
   apiVersion: v1
   metadata:
     name: test-pvc-project-1
     namespace: project-1
   annotations:
     trident.netapp.io/reclaimPolicy: Retain
   spec:
     accessModes:
     - ReadWriteOnce
     resources:
       requests:
       storage: 1Gi
   storageClassName: project-1-sc
   EOF
   ```
4. Check the PV associated with the PVC.

```sh
oc get pv
```

5. Validate that the PV and its volume is created in an SVM dedicated to project-1 on NetApp ONTAP.

```sh
volume show -vserver project-1-svm
```

6. Create a pod in project-1 and mount the PVC created in previous step.

```sh
cat << EOF | oc create -f -
kind: Pod
apiVersion: v1
metadata:
  name: test-pvc-pod
  namespace: project-1
spec:
  volumes:
    - name: test-pvc-project-1
      persistentVolumeClaim:
        claimName: test-pvc-project-1
  containers:
    - name: test-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: test-pvc-project-1
EOF
```

7. Check if the pod is running and whether it mounted the volume.

```sh
oc describe pods test-pvc-pod -n project-1
```

**Validate access to create PVCs or pods in another project or use resources dedicated to another project**

1. Log in as ocp-project-1-user, developer in project-1.
2. Create a PVC in project-1 using the storageclass that is assigned to project-2.
3. Create a PVC in project-2.

```yaml
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-2-sc-1
  namespace: project-2
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
    storageClassName: project-1-sc
EOF
```

4. Make sure that PVCs test-pvc-project-1-sc-2 and test-pvc-project-2-sc-1 were not created.

```bash
oc get pvc -n project-1
oc get pvc -n project-2
```

5. Create a pod in project-2.
Validate access to view and edit Projects, ResourceQuotas, and StorageClasses

1. Log in as ocp-project-1-user, developer in project-1.
2. Check access to create new projects.
   
   ```bash
   oc create ns sub-project-1
   ```

3. Validate access to view projects.

   ```bash
   oc get ns
   ```

4. Check if the user can view or edit ResourceQuotas in project-1.

   ```bash
   oc get resourcequotas -n project-1
   oc edit resourcequotas project-1-sc-rq -n project-1
   ```

5. Validate that the user has access to view the storageclasses.

   ```bash
   oc get sc
   ```

6. Check access to describe the storageclasses.

7. Validate the user’s access to edit the storageclasses.

   ```bash
   oc edit sc project-1-sc
   ```
Scaling: Adding more projects

In a multitenant configuration, adding new projects with storage resources requires additional configuration to make sure that multitenancy is not violated. For adding more projects in a multitenant cluster, complete the following steps:

1. Log into the NetApp ONTAP cluster as a storage admin.
2. Navigate to Storage → Storage VMs and click Add. Create a new SVM dedicated to project-3. Also create a vsadmin account to manage the SVM and its resources.
3. Log into the Red Hat OpenShift cluster as cluster admin.
4. Create a new project.
   
   oc create ns project-3

5. Make sure that the user group for project-3 is created on IdP and synchronized with the OpenShift cluster.
6. Create the developer role for project-3.

```bash
cat << EOF | oc create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: project-3
  name: developer-project-3
rules:
- verbs:
  - '*'
    apiGroups:
    - apps
    - batch
    - autoscaling
    - extensions
    - networking.k8s.io
    - policy
    - apps.openshift.io
    - build.openshift.io
    - image.openshift.io
    - ingress.operator.openshift.io
    - route.openshift.io
    - snapshot.storage.k8s.io
    - template.openshift.io
  resources:
  - '*'
- verbs:
  - '*'
    apiGroups:
    - ''
  resources:
  - bindings
  - configmaps
  - endpoints
  - events
  - persistentvolumeclaims
  - pods
  - pods/log
  - pods/attach
  - podtemplates
  - replicationcontrollers
  - services
EOF
```
The role definition provided in this section is just an example. The developer role must be defined based on the end-user requirements.

7. Create RoleBinding for developers in project-3 binding the developer-project-3 role to the corresponding group (ocp-project-3) in project-3.

```
cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-3-developer
  namespace: project-3
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-project-3
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: developer-project-3
EOF
```

8. Login to the Red Hat OpenShift cluster as storage admin

9. Create a Trident backend and map it to the SVM dedicated to project-3. NetApp recommends using the SVM’s vsadmin account to connect the backend to the SVM instead of using the ONTAP cluster administrator.
We are using the ontap-nas driver for this example. Use the appropriate driver for creating the backend based on the use-case.

We assume that Trident is installed in the trident project.

10. Create the storage class for project-3 and configure it to use the storage pools from backend dedicated to project-3.

```bash
cat << EOF | oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: project-3-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
  storagePools: "nfs_project_3::*"
EOF
```

11. Create a ResourceQuota to restrict resources in project-3 requesting storage from storageclasses dedicated to other projects.
cat << EOF | oc create -f -
kind: ResourceQuota
apiVersion: v1
metadata:
  name: project-3-sc-rq
  namespace: project-3
spec:
  hard:
    project-1-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
    project-2-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
EOF

12. Patch the ResourceQuotas in other projects to restrict resources in those projects from accessing storage from the storageclass dedicated to project-3.

oc patch resourcequotas project-1-sc-rq -n project-1 --patch '{"spec":{"hard":{"project-3-sc.storageclass.storage.k8s.io/persistentvolumeclaims": 0}}}'
oc patch resourcequotas project-2-sc-rq -n project-2 --patch '{"spec":{"hard":{"project-3-sc.storageclass.storage.k8s.io/persistentvolumeclaims": 0}}}'

Red Hat OpenShift Virtualization with NetApp ONTAP

Depending on the specific use case, both containers and virtual machines (VMs) can serve as optimal platforms for different types of applications. Therefore, many organizations run some of their workloads on containers and some on VMs. Often, this leads organizations to face additional challenges by having to manage separate platforms: a hypervisor for VMs and a container orchestrator for applications.

To address this challenge, Red Hat introduced OpenShift Virtualization (formerly known as Container Native Virtualization) starting from OpenShift version 4.6. The OpenShift Virtualization feature enables you to run and manage virtual machines alongside containers on the same OpenShift Container Platform installation, providing hybrid management capability to automate deployment and management of VMs through operators. In addition to creating VMs in OpenShift, with OpenShift Virtualization, Red Hat also supports importing VMs from VMware vSphere, Red Hat Virtualization, and Red Hat OpenStack Platform deployments.
Certain features like live VM migration, VM disk cloning, VM snapshots and so on are also supported by OpenShift Virtualization with assistance from Astra Trident when backed by NetApp ONTAP. Examples of each of these workflows are discussed later in this document in their respective sections.

To learn more about Red Hat OpenShift Virtualization, see the documentation here.

**Deployment for OpenShift Virtualization**

**Deploy Red Hat OpenShift Virtualization with NetApp ONTAP**

**Prerequisites**

- A Red Hat OpenShift cluster (later than version 4.6) installed on bare-metal infrastructure with RHCOS worker nodes
- The OpenShift cluster must be installed via installer provisioned infrastructure (IPI)
- Deploy Machine Health Checks to maintain HA for VMs
- A NetApp ONTAP cluster
- Astra Trident installed on the OpenShift cluster
- A Trident backend configured with an SVM on ONTAP cluster
- A StorageClass configured on the OpenShift cluster with Astra Trident as the provisioner
- Cluster-admin access to Red Hat OpenShift cluster
- Admin access to NetApp ONTAP cluster
- An admin workstation with tridentctl and oc tools installed and added to $PATH

Because OpenShift Virtualization is managed by an operator installed on the OpenShift cluster, it imposes additional overhead on memory, CPU, and storage, which must be accounted for while planning the hardware requirements for the cluster. See the documentation here for more details.

Optionally, you can also specify a subset of the OpenShift cluster nodes to host the OpenShift Virtualization operators, controllers, and VMs by configuring node placement rules. To configure node placement rules for OpenShift Virtualization, follow the documentation here.

For the storage backing OpenShift Virtualization, NetApp recommends having a dedicated StorageClass that
requests storage from a particular Trident backend, which in turn is backed by a dedicated SVM. This maintains a level of multitenancy with regard to the data being served for VM-based workloads on the OpenShift cluster.

**Deploy Red Hat OpenShift Virtualization with NetApp ONTAP**

To install OpenShift Virtualization, complete the following steps:

1. Log into the Red Hat OpenShift bare-metal cluster with cluster-admin access.
2. Select Administrator from the Perspective drop down.

4. Select the OpenShift Virtualization tile and click Install.
5. On the Install Operator screen, leave all default parameters and click Install.
6. Wait for the operator installation to complete.

7. After the operator has installed, click Create HyperConverged.

8. On the Create HyperConverged screen, click Create, accepting all default parameters. This step starts the installation of OpenShift Virtualization.
9. After all the pods move to the Running state in the openshift-cnv namespace and the OpenShift Virtualization operator is in the Succeeded state, the operator is ready to use. VMs can now be created on the OpenShift cluster.

Workflows

Workflows: Red Hat OpenShift Virtualization with NetApp ONTAP

Create VM

VMs are stateful deployments that require volumes to host the operating system and data. With CNV, because
the VMs are run as pods, the VMs are backed by PVs hosted on NetApp ONTAP through Trident. These volumes are attached as disks and store the entire filesystem including the boot source of the VM.

To create a virtual machine on the OpenShift cluster, complete the following steps:

1. Navigate to Workloads > Virtualization > Virtual Machines and click Create > With Wizard.
2. Select the desired the operating system and click Next.
3. If the selected operating system has no boot source configured, you must configure it. For Boot Source, select whether you want to import the OS image from an URL or from a registry and provide the corresponding details. Expand Advanced and select the Trident-backed StorageClass. Then click Next.
4. If the selected operating system already has a boot source configured, the previous step can be skipped.

5. In the Review and Create pane, select the project you want to create the VM in and furnish the VM details. Make sure that the boot source is selected to be Clone and boot from CD-ROM with the appropriate PVC assigned for the selected OS.

---

Boot source
This template does not have a boot source. Provide a custom boot source for this CentOS 8.0+ VM virtual machine.

Boot source type *
Import via URL (creates PVC)

Import URL *
https://access.cdn.redhat.com/content/origin/files/sha256/58/588167f828001e57688ec4b9b31c11a59d532489f527488ebc89ac5e952

Example: For RHEL, visit the RHEL download page (requires login) and copy the download link URL of the KVM guest image

Mount this as a CD-ROM boot source

Persistent Volume Claim size *
5 GiB

Ensure your PVC size covers the requirements of the uncompressed image and any other space requirements. More storage can be added later.

Advanced

Storage class *
basic (default)

Access mode *
Single User (RWO)

Volume mode *
Filesystem
6. If you wish to customize the virtual machine, click Customize Virtual Machine and modify the required parameters.

7. Click Create Virtual Machine to create the virtual machine; this spins up a corresponding pod in the background.

When a boot source is configured for a template or an operating system from an URL or from a registry, it creates a PVC in the openshift-virtualization-os-images project and downloads the KVM guest image to the PVC. You must make sure that template PVCs have enough provisioned space to accommodate the KVM guest image for the corresponding OS. These PVCs are then cloned and attached as rootdisks to virtual machines when they are created using the respective templates in any project.

**Workflows: Red Hat OpenShift Virtualization with NetApp ONTAP**

**VM Live Migration**

Live Migration is a process of migrating a VM instance from one node to another in an OpenShift cluster with no downtime. For live migration to work in an OpenShift cluster, VMs must be bound to PVCs with shared ReadWriteMany access mode. Astra Trident backend configured with an SVM on a NetApp ONTAP cluster that is enabled for NFS protocol supports shared ReadWriteMany access for PVCs. Therefore, the VMs with PVCs that are requested from StorageClasses provisioned by Trident from NFS-enabled SVM can be migrated with no downtime.
To create a VM bound to PVCs with shared ReadWriteMany access:

1. Navigate to Workloads > Virtualization > Virtual Machines and click Create > With Wizard.
2. Select the desired the operating system and click Next. Let us assume the selected OS already had a boot source configured with it.
3. In the Review and Create pane, select the project you want to create the VM in and furnish the VM details. Make sure that the boot source is selected to be Clone and boot from CD-ROM with the appropriate PVC assigned for the selected OS.
4. Click Customize Virtual Machine and then click Storage.
5. Click the ellipsis next to rootdisk, and make sure that the storageclass provisioned using Trident is selected. Expand Advanced and select Shared Access (RWX) for Access Mode. Then click Save.
6. Click Review and confirm and then click Create Virtual Machine.

To manually migrate a VM to another node in the OpenShift cluster, complete the following steps.

1. Navigate to Workloads > Virtualization > Virtual Machines.
2. For the VM you wish to migrate, click the ellipsis, and then click Migrate the Virtual Machine.
3. Click Migrate when the message pops up to confirm.

A VM instance in an OpenShift cluster automatically migrates to another node when the original node is placed into maintenance mode if the evictionStrategy is set to LiveMigrate.

**Workflows: Red Hat OpenShift Virtualization with NetApp ONTAP**

**VM cloning**

Cloning an existing VM in OpenShift is achieved with the support of Astra Trident’s Volume CSI cloning feature. CSI volume cloning allows for creation of a new PVC using an existing PVC as the data source by duplicating its PV. After the new PVC is created, it functions as a separate entity and without any link to or dependency on the source PVC.

There are certain restrictions with CSI volume cloning to consider:

1. Source PVC and destination PVC must be in the same project.
2. Cloning is supported within the same storage class.
3. Cloning can be performed only when source and destination volumes use the same VolumeMode setting; for example, a block volume can only be cloned to another block volume.
VMs in an OpenShift cluster can be cloned in two ways:

1. By shutting down the source VM
2. By keeping the source VM live

**By Shutting down the source VM**

Cloning an existing VM by shutting down the VM is a native OpenShift feature that is implemented with support from Astra Trident. Complete the following steps to clone a VM.

1. Navigate to Workloads > Virtualization > Virtual Machines and click the ellipsis next to the virtual machine you wish to clone.
2. Click Clone Virtual Machine and provide the details for the new VM.
3. Click Clone Virtual Machine; this shuts down the source VM and initiates the creation of the clone VM.

4. After this step is completed, you can access and verify the content of the cloned VM.
By keeping the source VM live

An existing VM can also be cloned by cloning the existing PVC of the source VM and then creating a new VM using the cloned PVC. This method does not require you to shut down the source VM. Complete the following steps to clone a VM without shutting it down.

1. Navigate to Storage > PersistentVolumeClaims and click the ellipsis next to the PVC that is attached to the source VM.
2. Click Clone PVC and furnish the details for the new PVC.

3. Then click Clone. This creates a PVC for the new VM.
4. Navigate to Workloads > Virtualization > Virtual Machines and click Create > With YAML.
5. In the spec > template > spec > volumes section, attach the cloned PVC instead of the container disk. Provide all other details for the new VM according to your requirements.
- name: rootdisk
  persistentVolumeClaim:
    claimName: rhel8-short-frog-rootdisk-28dgb-clone

6. Click Create to create the new VM.
7. After the VM is created successfully, access and verify that the new VM is a clone of the source VM.

**Workflows: Red Hat OpenShift Virtualization with NetApp ONTAP**

**Create VM from a Snapshot**

With Astra Trident and Red Hat OpenShift, users can take a snapshot of a persistent volume on Storage Classes provisioned by it. With this feature, users can take a point-in-time copy of a volume and use it to create a new volume or restore the same volume back to a previous state. This enables or supports a variety of use-cases, from rollback to clones to data restore.

For Snapshot operations in OpenShift, the resources VolumeSnapshotClass, VolumeSnapshot, and VolumeSnapshotContent must be defined.

- A VolumeSnapshotContent is the actual snapshot taken from a volume in the cluster. It is cluster-wide resource analogous to PersistentVolume for storage.
- A VolumeSnapshot is a request for creating the snapshot of a volume. It is analogous to a PersistentVolumeClaim.
- VolumeSnapshotClass lets the administrator specify different attributes for a VolumeSnapshot. It allows you to have different attributes for different snapshots taken from the same volume.
To create Snapshot of a VM, complete the following steps:

1. Create a VolumeSnapshotClass that can then be used to create a VolumeSnapshot. Navigate to Storage > VolumeSnapshotClasses and click Create VolumeSnapshotClass.

2. Enter the name of the Snapshot Class, enter csi.trident.netapp.io for the driver, and click Create.
3. Identify the PVC that is attached to the source VM and then create a Snapshot of that PVC. Navigate to Storage > VolumeSnapshots and click Create VolumeSnapshots.

4. Select the PVC that you want to create the Snapshot for, enter the name of the Snapshot or accept the default, and select the appropriate VolumeSnapshotClass. Then click Create.

5. This creates the snapshot of the PVC at that point in time.
Create a new VM from the snapshot

1. First, restore the Snapshot into a new PVC. Navigate to Storage > VolumeSnapshots, click the ellipsis next to the Snapshot that you wish to restore, and click Restore as new PVC.

2. Enter the details of the new PVC and click Restore. This creates a new PVC.

3. Next, create a new VM from this PVC. Navigate to Workloads > Virtualization > Virtual Machines and click Create > With YAML.

4. In the spec > template > spec > volumes section, specify the new PVC created from Snapshot instead of

---

## Restore as new PVC

When restore action for snapshot `rhel8-short-frog-rootdisk-28dvd-snapshot` is finished a new crash-consistent PVC copy will be created.

- **Name**: `rhel8-short-frog-rootdisk-28dvd-snapshot-restore`
- **Storage Class**: `basic`
- **Access Mode**: Single User (RWO)
- **Size**: 20 GiB

---

<table>
<thead>
<tr>
<th>Created at</th>
<th>Namespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 21, 12:46 am</td>
<td>default</td>
</tr>
</tbody>
</table>

**Status**: Ready

**API version**: `snapshot.storage.k8s.io/v1`

**Size**: 20 GiB
from the container disk. Provide all other details for the new VM according to your requirements.

```yaml
- name: rootdisk
  persistentVolumeClaim:
    claimName: rhel8-short-frog-rootdisk-28dvd-snapshot-restore
```

5. Click Create to create the new VM.

6. After the VM is created successfully, access and verify that the new VM has the same state as that of the VM whose PVC was used to create the snapshot at the time when the snapshot was created.

**Workflows: Red Hat OpenShift Virtualization with NetApp ONTAP**

**Migration of VM from VMware to OpenShift Virtualization using Migration Toolkit for Virtualization**

In this section, we will see how to use the Migration Toolkit for Virtualization (MTV) to migrate virtual machines from VMware to OpenShift Virtualization running on OpenShift Container platform and integrated with NetApp ONTAP storage using Astra Trident.

The following video shows a demonstration of the migration of a RHEL VM from VMware to OpenShift Virtualization using ontap-san for persistent storage.

**Using Red Hat MTV to migrate VMs to OpenShift Virtualization with NetApp ONTAP Storage**

The following diagram shows a high level view of the migration of a VM from VMware to Red Hat OpenShift Virtualization.

---

**Prerequisites for the sample migration**
On VMware

- A RHEL 9 VM using rhel 9.3 with the following configurations were installed:
  - CPU: 2, Memory: 20 GB, Hard disk: 20 GB
  - user credentials: root user and an admin user credentials
- After the VM was ready, postgresql server was installed.
  - postgresql server was started and enabled to start on boot

```
systemctl start postgresql.service
systemctl enable postgresql.service
The above command ensures that the server can start in the VM in OpenShift Virtualization after migration
```

- Added 2 databases, 1 table and 1 row in the table were added. Refer here for the instructions for installing postgresql server on RHEL and creating database and table entries.

Ensure that you start the postgresql server and enable the service to start at boot.

On OpenShift Cluster

The following installations were completed before installing MTV:

- OpenShift Cluster 4.13.34
- Astra Trident 23.10
- Multipath on the cluster nodes enabled for iSCSI (for ontap-san storage class). See the provided yaml to create a daemon set that enables iSCSI on each node in the cluster.
- Trident backend and Storage class for ontap SAN using iSCSI. See the provided yaml files for trident backend and storage class.
- OpenShift Virtualization

To install iscsi and multipath on the OpenShift Cluster nodes use the yaml file given below

```
Preparing the cluster nodes for iSCSI

apiVersion: apps/v1
kind: DaemonSet
metadata:
  namespace: trident
  name: trident-iscsi-init
labels:
  name: trident-iscsi-init
spec:
  selector:
    matchLabels:
      name: trident-iscsi-init
```
metadata:
  labels:
    name: trident-iscsi-init
spec:
  hostNetwork: true
  serviceAccount: trident-node-linux
  initContainers:
    - name: init-node
      command:
        - nsenter
        - --mount=/proc/1/ns/mnt
        - --
        - sh
        - -c
      args: ["${STARTUP_SCRIPT}" ]
  image: alpine:3.7
  env:
    - name: STARTUP_SCRIPT
      value: |
        #!/bin/bash
        sudo yum install -y lsscsi iscsi-initiator-utils sg3_utils
device-mapper-multipath
        rpm -q iscsi-initiator-utils
        sudo sed -i 's/\(node.session.scan\)/\1 = manual/'
        /etc/iscsi/iscsid.conf
        cat /etc/iscsi/ initiatorname. iscsi
        sudo mpathconf -- enable --with_multipathd y --find_multipaths
        n
        sudo systemctl enable --now iscsid multipathd
        sudo systemctl enable --now iscsi
  securityContext:
    privileged: true
  hostPID: true
  containers:
    - name: wait
      image: k8s.gcr.io/pause:3.1
  hostNetwork: true
  tolerations:
    - effect: NoSchedule
      key: node-role.kubernetes.io/master
  updateStrategy:
    type: RollingUpdate

Use the following yaml file to create trident backend configuration for using ontap san storage

Trident backend for iSCSI
Use the following yaml file to create trident storage class configuration for using ontap san storage

**Trident storage class for iSCSI**

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
  allowVolumeExpansion: true
```

**Install MTV**

Now you can install the Migration Toolkit for virtualization (MTV). Refer to the instructions provided [here](#) for help with the installation.

The Migration Toolkit for Virtualization (MTV) user interface is integrated into the OpenShift web console. You can refer [here](#) to start using the user interface for various tasks.
Create Source Provider

In order to migrate the RHEL VM from VMware to OpenShift Virtualization, you need to first create the source provider for VMware. Refer to the instructions here to create the source provider.

You need the following to create your VMware source provider:

- VCenter url
- VCenter Credentials
- VCenter server thumbprint
- VDDK image in a repository

Sample source provider creation:
The Migration Toolkit for Virtualization (MTV) uses the VMware Virtual Disk Development Kit (VDDK) SDK to accelerate transferring virtual disks from VMware vSphere. Therefore, creating a VDDK image, although optional, is highly recommended.

To make use of this feature, you download the VMware Virtual Disk Development Kit (VDDK), build a VDDK image, and push the VDDK image to your image registry.

Follow the instructions provided here to create and push the VDDK image to a registry accessible from the OpenShift Cluster.

Create Destination provider

The host cluster is automatically added as the OpenShift virtualization provider is the source provider.

Create Migration Plan

Follow the instructions provided here to create a migration plan.

While creating a plan, you need to create the following if not already created:

- A network mapping to map the source network to the target network.
- A storage mapping to map the source datastore to the target storage class. For this you can choose ontap-san storage class.

Once the migration plan is created, the status of the plan should show Ready and you should now be able to Start the plan.

Clicking on Start will run through a sequence of steps to complete the migration of the VM.
When all steps are completed, you can see the migrated VMs by clicking on the virtual machines under Virtualization in the left-side navigation menu.

Instructions to access the virtual machines are provided here.

You can log into the virtual machine and verify the contents of the postgresql databases. The databases, tables and the entries in the table should be the same as what was created on the source VM.

Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp

Advanced Cluster Management for Kubernetes: Red Hat OpenShift with NetApp

As a containerized application transitions from development to production, many organizations require multiple Red Hat OpenShift clusters to support the testing and deployment of that application. In conjunction with this, organizations usually host multiple applications or workloads on OpenShift clusters. Therefore, each organization ends up managing a set of clusters, and OpenShift administrators must thus face the added challenge of managing and maintaining multiple clusters across a range of environments that span multiple on-premises data centers and public clouds. To address these challenges, Red Hat introduced Advanced Cluster Management for Kubernetes.

Red Hat Advanced Cluster Management for Kubernetes enables you to perform the following tasks:

1. Create, import, and manage multiple clusters across data centers and public clouds
2. Deploy and manage applications or workloads on multiple clusters from a single console
3. Monitor and analyze health and status of different cluster resources
4. Monitor and enforce security compliance across multiple clusters

Red Hat Advanced Cluster Management for Kubernetes is installed as an add-on to a Red Hat OpenShift cluster, and it uses this cluster as a central controller for all its operations. This cluster is known as hub cluster, and it exposes a management plane for the users to connect to Advanced Cluster Management. All the other OpenShift clusters that are either imported or created via the Advanced Cluster Management console are
managed by the hub cluster and are called managed clusters. It installs an agent called Klusterlet on the managed clusters to connect them to the hub cluster and serve the requests for different activities related to cluster lifecycle management, application lifecycle management, observability, and security compliance.

Deployment

Deploy Advanced Cluster Management for Kubernetes

Prerequisites

1. A Red Hat OpenShift cluster (greater than version 4.5) for the hub cluster
2. Red Hat OpenShift clusters (greater than version 4.4.3) for managed clusters
3. Cluster-admin access to the Red Hat OpenShift cluster
4. A Red Hat subscription for Advanced Cluster Management for Kubernetes

Advanced Cluster Management is an add-on on for the OpenShift cluster, so there are certain requirements and restrictions on the hardware resources based on the features used across the hub and managed clusters. You need to take these issues into account when sizing the clusters. See the documentation here for more details.

Optionally, if the hub cluster has dedicated nodes for hosting infrastructure components and you would like to install Advanced Cluster Management resources only on those nodes, you need to add tolerations and selectors to those nodes accordingly. For more details, see the documentation here.

Deploy Advanced Cluster Management for Kubernetes

To install Advanced Cluster Management for Kubernetes on an OpenShift cluster, complete the following steps:

1. Choose an OpenShift cluster as the hub cluster and log into it with cluster-admin privileges.

For more information, see the documentation here.

**Advanced Cluster Management for Kubernetes**

2.23 provided by Red Hat

![Advanced Cluster Management for Kubernetes](image)

**Latest version**
2.2.3

**Capability level**
- Basic Install
- Seamless Upgrades
- Full Lifecycle
- Deep Insights
- Auto Pilot

**Provider type**
Red Hat

**Provider**
Red Hat

**Infrastructure features**
Disconnected

Red Hat Advanced Cluster Management for Kubernetes provides the multicloud hub, a central management console for managing multiple Kubernetes-based clusters across data centers, public clouds, and private clouds. You can use the hub to create Red Hat OpenShift Container Platform clusters on selected providers, or import existing Kubernetes-based clusters. After the clusters are managed, you can set compliance requirements to ensure that the clusters maintain the specified security requirements. You can also deploy business applications across your clusters.

Red Hat Advanced Cluster Management for Kubernetes also provides the following operators:

- Multicluster subscriptions: An operator that provides application management capabilities including subscribing to resources from a channel and deploying those resources on MCH-managed Kubernetes clusters based on placement rules.
- Hive for Red Hat OpenShift: An operator that provides APIs for provisioning and performing initial configuration of OpenShift clusters. These operators are used by the multicloud hub to provide its provisioning and application-management capabilities.

**How to Install**

Use of this Red Hat product requires a licensing and subscription agreement.

4. On the Install Operator screen, provide the necessary details (NetApp recommends retaining the default parameters) and click Install.
Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

- **Update channel**
  - release-2.0
  - release-2.1
  - release-2.2

- **Installation mode**
  - All namespaces on the cluster (default)
    - This mode is not supported by this Operator
  - A specific namespace on the cluster
    - Operator will be available in a single Namespace only

- **Installed Namespace**
  - Operator recommended Namespace: open-cluster-management

  - **Namespace creation**
    - Namespace open-cluster-management does not exist and will be created.

- **Approval strategy**
  - Automatic
  - Manual

5. Wait for the operator installation to complete.

6. After the operator is installed, click Create MultiClusterHub.
7. On the Create MultiClusterHub screen, click Create after furnishing the details. This initiates the installation of a multi-cluster hub.

8. After all the pods move to the Running state in the open-cluster-management namespace and the operator moves to the Succeeded state, Advanced Cluster Management for Kubernetes is installed.
9. It takes some time to complete the hub installation, and, after it is done, the MultiCluster hub moves to Running state.

10. It creates a route in the open-cluster-management namespace. Connect to the URL in the route to access the Advanced Cluster Management console.
Features: Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp

Cluster Lifecycle Management

To manage different OpenShift clusters, you can either create or import them into Advanced Cluster Management.

1. First navigate to Automate Infrastructures > Clusters.
2. To create a new OpenShift cluster, complete the following steps:
   a. Create a provider connection: Navigate to Provider Connections and click Add a Connection, provide all the details corresponding to the selected provider type and click Add.
   b. To create a new cluster, navigate to Clusters and click Add a Cluster > Create a Cluster. Provide the details for the cluster and the corresponding provider and click Create.
c. After the cluster is created, it appears in the cluster list with the status Ready.

3. To import an existing cluster, complete the following steps:
   a. Navigate to Clusters and click Add a Cluster > Import an Existing Cluster.
   b. Enter the name of the cluster and click Save Import and Generate Code. A command to add the existing cluster is displayed.
   c. Click Copy Command and run the command on the cluster to be added to the hub cluster. This initiates the installation of the necessary agents on the cluster, and, after this process is complete, the cluster appears in the cluster list with status Ready.
4. After you create and import multiple clusters, you can monitor and manage them from a single console.

**Features: Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp**

**Application lifecycle management**

To create an application and manage it across a set of clusters,

1. Navigate to Manage Applications from the sidebar and click Create Application. Provide the details of the application you would like to create and click Save.
2. After the application components are installed, the application appears in the list.

3. The application can now be monitored and managed from the console.
Governance and risk

This feature allows you to define the compliance policies for different clusters and make sure that the clusters adhere to it. You can configure the policies to either inform or remediate any deviations or violations of the rules.

1. Navigate to Governance and Risk from the sidebar.

2. To create compliance policies, click Create Policy, enter the details of the policy standards, and select the clusters that should adhere to this policy. If you want to automatically remediate the violations of this policy, select the checkbox Enforce if Supported and click Create.
3. After all the required policies are configured, any policy or cluster violations can be monitored and remediated from Advanced Cluster Management.
Features: Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp

Observability

Advanced Cluster Management for Kubernetes provides a way to monitor the nodes, pods, and applications, and workloads across all the clusters.

1. Navigate to Observe Environments > Overview.
2. All pods and workloads across all clusters are monitored and sorted based on a variety of filters. Click Pods to view the corresponding data.

3. All nodes across the clusters are monitored and analyzed based on a variety of data points. Click Nodes to get more insight into the corresponding details.
4. All clusters are monitored and organized based on different cluster resources and parameters. Click Clusters to view cluster details.

Features: Advanced Cluster Management for Kubernetes on Red Hat OpenShift with NetApp

Create resources on multiple clusters

Advanced Cluster Management for Kubernetes allows users to create resources on one or more managed clusters simultaneously from the console. As an example, if you have OpenShift clusters at different sites backed with different NetApp ONTAP clusters and want to provision PVC’s at both sites, you can click the (+) sign on the top bar. Then select the clusters on which you want to create the PVC, paste the resource YAML, and click Create.
Videos and Demos: Red Hat OpenShift with NetApp

The following videos demonstrate some of the capabilities documented in this document:

Using Red Hat MTV to migrate VMs to OpenShift Virtualization with NetApp ONTAP Storage

Accelerate Software Development with Astra Control and NetApp FlexClone Technology - Red Hat OpenShift with NetApp

Leverage NetApp Astra Control to Perform Post-mortem Analysis and Restore Your Application

Data Protection in CI/CD pipeline with Astra Control Center

Workload Migration using Astra Control Center - Red Hat OpenShift with NetApp

Workload Migration - Red Hat OpenShift with NetApp

Installing OpenShift Virtualization - Red Hat OpenShift with NetApp

Deploying a Virtual Machine with OpenShift Virtualization - Red Hat OpenShift with NetApp

NetApp HCI for Red Hat OpenShift on Red Hat Virtualization

Additional Information: Red Hat OpenShift with NetApp

To learn more about the information described in this document, review the following websites:

• NetApp Documentation
  https://docs.netapp.com/
This reference document provides deployment validation of different flavors of VMware Tanzu Kubernetes solutions, deployed either as Tanzu Kubernetes Grid (TKG), Tanzu Kubernetes Grid Service (TKGS), or Tanzu Kubernetes Grid Integrated (TKGI) in several different data center environments as validated by NetApp. It also describes storage integration with NetApp storage systems and the Astra Trident storage orchestrator for the management of persistent storage and Astra Control Center for the backup and cloning of the stateful applications using that persistent storage. Lastly, the document provides video demonstrations of the solution integrations and validations.

Use cases

The VMware Tanzu with NetApp solution is architected to deliver exceptional value for customers with the following use cases:

- Easy to deploy and manage VMware Tanzu Kubernetes Grid offerings deployed on VMware vSphere and integrated with NetApp storage systems.
- The combined power of enterprise container and virtualized workloads with VMware Tanzu Kubernetes Grid offerings.
- Real world configuration and use cases highlighting the features of VMware Tanzu when used with NetApp storage and the NetApp Astra suite of products.
- Application-consistent protection or migration of containerized workloads deployed on VMware Tanzu Kubernetes Grid clusters whose data resides on NetApp storage systems using Astra Control Center.
Business value

Enterprises are increasingly adopting DevOps practices to create new products, shorten release cycles, and rapidly add new features. Because of their innate agile nature, containers and microservices play a crucial role in supporting DevOps practices. However, practicing DevOps at a production scale in an enterprise environment presents its own challenges and imposes certain requirements on the underlying infrastructure, such as the following:

- High availability at all layers in the stack
- Ease of deployment procedures
- Non-disruptive operations and upgrades
- API-driven and programmable infrastructure to keep up with microservices agility
- Multitenancy with performance guarantees
- Ability to run virtualized and containerized workloads simultaneously
- Ability to scale infrastructure independently based on workload demands
- Ability to deploy in a hybrid-cloud model with containers running in both on-premises data centers as well as in the cloud.

VMware Tanzu with NetApp acknowledges these challenges and presents a solution that helps address each concern by deploying VMware Tanzu Kubernetes offerings in the customer’s choice of hybrid cloud environment.

Technology overview

The VMware Tanzu with NetApp solution is comprised of the following major components:

**VMware Tanzu Kubernetes platforms**

VMware Tanzu comes in a variety of flavors that the solutions engineering team at NetApp has validated in our labs. Each Tanzu release successfully integrates with the NetApp storage portfolio, and each can help meet certain infrastructure demands. The following bulleted highlights describe the features and offerings of each version of Tanzu described in this document.

**VMware Tanzu Kubernetes Grid (TKG)**

- Standard upstream Kubernetes environment deployed in a VMware vSphere environment.
- Formerly known as Essential PKS (from Heptio acquisition, Feb 2019).
- TKG is deployed with a separate management cluster instance for support on vSphere 6.7U3 onward.
- TKG deployments can be deployed in the cloud as well with AWS or Azure.
- Allows for use of Windows or Linux worker nodes (Ubuntu/Photon).
- NSX-T, HA Proxy, AVI networking, or load balancers can be used for control plane.
- TKG supports MetalLB for the application/data plane.
- Can use vSphere CSI as well as third party CSIs like NetApp Astra Trident.

**VMware Tanzu Kubernetes Grid Service (TKGS)**

- Standard upstream Kubernetes environment deployed in a VMware vSphere environment.
• Formerly known as Essential PKS (from Heptio acquisition, Feb 2019).
• TKGS deployed with supervisor cluster and workload clusters only on vSphere 7.0U1 onward.
• Allows for use of Windows or Linux worker nodes (Ubuntu/Photon).
• NSX-T, HA Proxy, AVI networking, or load balancers can be used for control plane.
• TKGS supports MetaLB for application/data plane.
• Can use vSphere CSI as well as third party CSIs like NetApp Astra Trident.
• Provides support for vSphere Pods with Tanzu, allowing pods to run directly on enabled ESXi hosts in the environment.

VMWare Tanzu Kubernetes Grid Integrated (TKGI)

• Formerly known as Enterprise PKS (from Heptio acquisition, Feb 2019).
• Can use NSX-T, HA Proxy, or Avi. You can also provide your own load balancer.
• Supported from vSphere 6.7U3 onward, as well as AWS, Azure, and GCP.
• Setup via wizard to allow for ease of deployment.
• Runs Tanzu in controlled immutable VMs managed by BOSH.
• Can make use vSphere CSI as well as third party CSIs like NetApp Astra Trident (some conditions apply).

vSphere with Tanzu (vSphere Pods)

• vSphere-native pods run in a thin, photon-based layer with prescribed virtual hardware for complete isolation.
• Requires NSX-T, but that allows for additional feature support such as a Harbor image registry.
• Deployed and managed in vSphere 7.0U1 onward using a virtual Supervisor cluster like TKGS. Runs pods directly on ESXi nodes.
• Fully vSphere integrated, highest visibility and control by vSphere administration.
• Isolated CRX-based pods for the highest level of security.
• Only supports vSphere CSI for persistent storage. No third-party storage orchestrators supported.

NetApp storage systems

NetApp has several storage systems perfect for enterprise data centers and hybrid cloud deployments. The NetApp portfolio includes NetApp ONTAP, NetApp Element, and NetApp e-Series storage systems, all of which can provide persistent storage for containerized applications.

For more information, visit the NetApp website here.

NetApp storage integrations

NetApp Astra Control Center offers a rich set of storage and application-aware data management services for stateful Kubernetes workloads, deployed in an on-prem environment, and powered by trusted NetApp data protection technology.

For more information, visit the NetApp Astra website here.

Astra Trident is an open-source, fully-supported storage orchestrator for containers and Kubernetes distributions, including VMware Tanzu.
For more information, visit the Astra Trident website here.

**Current support matrix for validated releases**

<table>
<thead>
<tr>
<th>Technology</th>
<th>Purpose</th>
<th>Software version</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetApp ONTAP</td>
<td>Storage</td>
<td>9.9.1</td>
</tr>
<tr>
<td>NetApp Astra Control Center</td>
<td>Application Aware Data Management</td>
<td>22.04</td>
</tr>
<tr>
<td>NetApp Astra Trident</td>
<td>Storage Orchestration</td>
<td>22.04.0</td>
</tr>
<tr>
<td>VMware Tanzu Kubernetes Grid</td>
<td>Container orchestration</td>
<td>1.4+</td>
</tr>
<tr>
<td>VMware Tanzu Kubernetes Grid Service</td>
<td>Container orchestration</td>
<td>0.0.15 [vSphere Namespaces]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.22.6 [Supervisor Cluster Kubernetes]</td>
</tr>
<tr>
<td>VMware Tanzu Kubernetes Grid Integrated</td>
<td>Container orchestration</td>
<td>1.13.3</td>
</tr>
<tr>
<td>VMware vSphere</td>
<td>Data center virtualization</td>
<td>7.0U3</td>
</tr>
<tr>
<td>VMware NSX-T Data Center</td>
<td>Networking and Security</td>
<td>3.1.3</td>
</tr>
<tr>
<td>VMware NSX Advanced Load Balancer</td>
<td>Load Balancer</td>
<td>20.1.3</td>
</tr>
</tbody>
</table>

**VMware Tanzu overview**

VMware Tanzu is a portfolio of products that enables enterprises to modernize their applications and the infrastructure they run on. VMware Tanzu's full stack of capabilities unites the development and IT operations teams on a single platform to embrace modernization in both their applications and their infrastructure consistently across on-premises and hybrid cloud environments to continuously deliver better software to production.
To understand more about the different offerings and their capabilities in the Tanzu portfolio, visit the documentation [here](#).

Regarding Tanzu’s Kubernetes Operations catalog, VMware has a variety of implementations for Tanzu Kubernetes Grid, all of which provision and manage the lifecycle of Tanzu Kubernetes clusters on a variety of platforms. A Tanzu Kubernetes cluster is a full-fledged Kubernetes distribution that is built and supported by VMware.

NetApp has tested and validated the deployment and interoperability of the following products from the VMware Tanzu portfolio in its labs:

- VMware Tanzu Kubernetes Grid (TKG)
- VMware Tanzu Kubernetes Grid Service (TKGS)
- VMware Tanzu Kubernetes Grid Integrated (TKGI)
- VMware vSphere with Tanzu (vSphere Pods)

**VMware Tanzu Kubernetes Grid (TKG) overview**

VMware Tanzu Kubernetes Grid, also known as TKG, lets you deploy Tanzu Kubernetes clusters across hybrid cloud or public cloud environments. TKG is installed as a management cluster, which is a Kubernetes cluster itself, that deploys and operates the Tanzu Kubernetes clusters. These Tanzu Kubernetes clusters are the workload Kubernetes clusters on which the actual workload is deployed.

Tanzu Kubernetes Grid builds on a few of the promising upstream community projects and delivers a Kubernetes platform that is developed, marketed, and supported by VMware. In addition to Kubernetes distribution, Tanzu Kubernetes Grid provides additional add-ons that are essential production-grade services such as registry, load balancing, authentication, and so on. VMware TKG with management cluster is widely used in vSphere 6.7 environments, and, even though it is supported, it is not a recommended deployment for vSphere 7 environments because TKGS has native integration capabilities with vSphere 7.
For more information on Tanzu Kubernetes Grid, refer to the documentation here.

Depending on whether the Tanzu Kubernetes Grid is being installed on-premises on vSphere cluster or in cloud environments, prepare and deploy Tanzu Kubernetes Grid by following the installation guide here.

After you have installed the management cluster for Tanzu Kubernetes Grid, deploy the user clusters or workload clusters as needed by following the documentation here. VMware TKG management cluster requires that an SSH key be provided for installation and operation of Tanzu Kubernetes clusters. This key can be used to log into the cluster nodes using the capv user.

**VMware Tanzu Kubernetes Grid Service (TKGS) overview**

VMware Tanzu Kubernetes Grid Service (also known as vSphere with Tanzu) lets you create and operate Tanzu Kubernetes clusters natively in vSphere and also allows you to run some smaller workloads directly on the ESXi hosts. It allows you to transform vSphere into a platform for running containerized workloads natively on the hypervisor layer. Tanzu Kubernetes Grid Service deploys a supervisor cluster on vSphere when enabled that deploys and operates the clusters required for the workloads. It is natively integrated with vSphere 7 and leverages many reliable vSphere features like vCenter SSO, Content Library, vSphere networking, vSphere storage, vSphere HA and DRS, and vSphere security for a more seamless Kubernetes experience.

vSphere with Tanzu offers a single platform for hybrid application environments where you can run your application components either in containers or in VMs, thus providing better visibility and ease of operations for developers, DevOps engineers, and vSphere administrators. VMware TKGS is only supported with vSphere 7 environments and is the only offering in Tanzu Kubernetes operations portfolio that allows you to run pods directly on ESXi hosts.
For more information on Tanzu Kubernetes Grid Service, follow the documentation here.

There are a lot of architectural considerations regarding feature sets, networking, and so on. Depending on the architecture chosen, the prerequisites and the deployment process of Tanzu Kubernetes Grid Service differ. To deploy and configure Tanzu Kubernetes Grid Service in your environment, follow the guide here. Furthermore, to log into the Tanzu Kubernetes cluster nodes deployed via TKGS, follow the procedure laid out in this link.

NetApp recommends that all the production environments be deployed in multiple master deployments for fault tolerance with the choice of worker nodes’ configuration to meet the requirements of the intended workloads. Thus, a recommended VM class for a highly intensive workload would have at least four vCPUs and 12GB of RAM.

When Tanzu Kubernetes clusters are created in a namespace, users with owner or edit permission can create pods directly in any namespace by using the user account. This is because users with the owner or edit permission are allotted the cluster administrator role. However, when creating deployments, daemon sets, stateful sets, or others in any namespace, you must assign a role with the required permissions to the corresponding service accounts. This is required because the deployments or daemon sets utilize service accounts to deploy the pods.

See the following example of ClusterRoleBinding to assign the cluster administrator role to all service accounts in the cluster:
VMware Tanzu Kubernetes Grid Integrated Edition (TKGI) overview

VMware Tanzu Kubernetes Grid Integrated (TKGI) Edition, formerly known as VMware Enterprise PKS, is a standalone container orchestration platform based on Kubernetes with capabilities such as life cycle management, cluster health monitoring, advanced networking, a container registry, and so on. TKGI provisions and manages Kubernetes clusters with the TKGI control plane, which consists of BOSH and Ops Manager.

TKGI can be installed and operated either on vSphere or OpenStack environments on-premises or in any of the major public clouds on their respective IaaS offerings. Furthermore, the integration of TKGI with NSX-T and Harbour enables wider use cases for enterprise workloads. To know more about TKGI and its capabilities, visit the documentation [here](#).

TKGI is installed in a variety of configurations on a variety of platforms based on different use-cases and designs. Follow the guide [here](#) to install and configure TKGI and its prerequisites. TKGI uses Bosh VMs as nodes for Tanzu Kubernetes clusters which run immutable configuration images and any manual changes on Bosh VMs do not remain persistent across reboots.

Important notes:
• NetApp Trident requires privileged container access. So, during TKGI installation, make sure to select the Enable Privileged Containers checkbox in the step to configure Tanzu Kubernetes cluster node plans.

- **Worker Node Instances**: 3
- **Worker Persistent Disk Size**: 50 GB
- **Worker VM Type**: medium.disk (cpu: 2, ram: 4 GB, disk: 32 GB)
- **Max Worker Node Instances**: 50
- **Enable Privileged Containers (Use with caution)**
- **Node Drain Timeout (minutes, min: 0, max: 1440)**: 0
- **Pod Shutdown Grace Period (seconds, min: -1, max: 86400)**: 10

• NetApp recommends that all production environments be deployed in multiple master deployments for fault tolerance with the choice of worker nodes' configuration to meet the requirements of the intended workloads. Thus, a recommended TKGI cluster plan would consist of at least three masters and three workers with at least four vCPUs and 12GB of RAM for a highly intensive workload.

**NetApp storage systems overview**

Unresolved directive in containers/vtwn_overview_netapp.adoc - include::/.../_include/containers_common_intro_sections.adoc[tags=netapp_overview_page;!netapp_overview_page_element]

**NetApp ONTAP**

Unresolved directive in containers/vtwn_netapp_ontap.adoc - include::/.../_include/containers_common_intro_sections.adoc[tags=netapp_ontap_page]

**NetApp storage integration overview**

Unresolved directive in containers/vtwn_overview_storint.adoc - include::/.../_include/containers_common_intro_sections.adoc[tags=storage_integration_overview]

**NetApp Astra Control overview**

Unresolved directive in containers/vtwn_overview_astra.adoc - include::/.../_include/containers_common_intro_sections.adoc[tags=astra_cc_overview]
Astra Control Center automation

Astra Control Center has a fully functional REST API for programmatic access. Users can use any programming language or utility to interact with Astra Control REST API endpoints. To learn more about this API, see the documentation here.

If you are looking for a ready-made software development toolkit for interacting with Astra Control REST APIs, NetApp provides a toolkit with the Astra Control Python SDK that you can download here.

If programming is not appropriate for your situation and you would like to use a configuration management tool, you can clone and run the Ansible playbooks that NetApp publishes here.

Astra Control Center installation prerequisites

Astra Control Center installation requires the following prerequisites:

- One or more Tanzu Kubernetes clusters, managed either by a management cluster or TKGS or TKGI. TKG workload clusters 1.4+ and TKGI user clusters 1.12.2+ are supported.
- Astra Trident must already be installed and configured on each of the Tanzu Kubernetes clusters.
- One or more NetApp ONTAP storage systems running ONTAP 9.5 or greater.

It’s a best practice for each Tanzu Kubernetes install at a site to have a dedicated SVM for persistent storage. Multi-site deployments require additional storage systems.

- A Trident storage backend must be configured on each Tanzu Kubernetes cluster with an SVM backed by an ONTAP cluster.
- A default StorageClass configured on each Tanzu Kubernetes cluster with Astra Trident as the storage provisioner.
- A load balancer must be installed and configured on each Tanzu Kubernetes cluster for load balancing and exposing Astra Control Center if you are using ingressType AccTraefik.
- An ingress controller must be installed and configured on each Tanzu Kubernetes cluster for exposing Astra Control Center if you are using ingressType Generic.
- A private image registry must be configured to host the NetApp Astra Control Center images.
- You must have Cluster Admin access to the Tanzu Kubernetes cluster where Astra Control Center is being installed.
- You must have Admin access to NetApp ONTAP clusters.
- A RHEL or Ubuntu admin workstation.

Install Astra Control Center

This solution describes an automated procedure for installing Astra Control Center using Ansible playbooks. If you are looking for a manual procedure to install Astra Control Center, follow the detailed installation and operations guide here.

Unresolved directive in containers/vtwn_overview_astra.adoc - include::../../_include/containers_astra_cc_install_ansible.adoc[Install Astra Control Center using Ansible]

Post Install Steps

1. It might take several minutes for the installation to complete. Verify that all the pods and services in the
netapp-astra-cc namespace are up and running.

[netapp-user@rhel7 ~]$ kubectl get all -n netapp-astra-cc

2. Check the acc-operator-controller-manager logs to ensure that the installation is completed.

[netapp-user@rhel7 ~]$ kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-operator -c manager -f

The following message indicates the successful installation of Astra Control Center.

{"level":"info","ts":1624054318.029971,"logger":"controllers.AstraControlCenter","msg":"Successfully Reconciled AstraControlCenter in [seconds]s","AstraControlCenter":"netapp-astra-cc/astra","ae.Version":"[22.04.0]"}

3. The username for logging into Astra Control Center is the email address of the administrator provided in the CRD file and the password is a string ACC- appended to the Astra Control Center UUID. Run the following command:

[netapp-user@rhel7 ~]$ oc get astracontrolcenters -n netapp-astra-cc

NAME     UUID
astra    345c55a5-bf2e-21f0-84b8-b6f2bce5e95f

In this example, the password is ACC-345c55a5-bf2e-21f0-84b8-b6f2bce5e95f.

4. Get the traefik service load balancer IP if the ingressType is AccTraefik.

[netapp-user@rhel7 ~]$ oc get svc -n netapp-astra-cc | egrep 'EXTERNAL|traefik'

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLUSTER-IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>traefik</td>
<td>LoadBalancer</td>
<td>172.30.99.142</td>
</tr>
<tr>
<td></td>
<td>PORT(S)</td>
<td>10.61.186.181 80:30343/TCP,443:30060/TCP</td>
</tr>
<tr>
<td></td>
<td>AGE</td>
<td>16m</td>
</tr>
</tbody>
</table>

5. Add an entry in the DNS server pointing the FQDN provided in the Astra Control Center CRD file to the EXTERNAL-IP of the traefik service.
6. Log into the Astra Control Center GUI by browsing its FQDN.
7. When you log into Astra Control Center GUI for the first time using the admin email address provided in CRD, you need to change the password.

8. If you wish to add a user to Astra Control Center, navigate to Account > Users, click Add, enter the details of the user, and click Add.
9. Astra Control Center requires a license for all of its functionalities to work. To add a license, navigate to Account > License, click Add License, and upload the license file.

If you encounter issues with the install or configuration of NetApp Astra Control Center, the knowledge base of known issues is available here.

Register your VMware Tanzu Kubernetes Clusters with the Astra Control Center

To enable the Astra Control Center to manage your workloads, you must first register your Tanzu Kubernetes clusters.

Register VMware Tanzu Kubernetes clusters

1. The first step is to add the Tanzu Kubernetes clusters to the Astra Control Center and manage them. Go to
Clusters and click Add a Cluster, upload the kubeconfig file for the Tanzu Kubernetes cluster, and click Select Storage.

2. Astra Control Center detects the eligible storage classes. Now select the way that storageclass provisions volumes using Trident backed by an SVM on NetApp ONTAP and click Review. In the next pane, verify the details and click Add Cluster.

3. When the cluster is added, it moves to the Discovering status while Astra Control Center inspects it and installs the necessary agents. The cluster status changes to Healthy after it is successfully registered.

4. Import ONTAP clusters as storage resources to be managed as backends by Astra Control Center. When Tanzu Kubernetes clusters are added to Astra and a storageclass is configured, it automatically discovers and inspects the ONTAP cluster backing the storageclass but does not import it into the Astra Control Center to be managed.

All Tanzu Kubernetes clusters to be managed by Astra Control Center should have access to the image registry that was used for its installation as the agents installed on the managed clusters pull the images from that registry.
5. To import the ONTAP clusters, navigate to Backends, click the dropdown, and select Manage next to the ONTAP cluster to be managed. Enter the ONTAP cluster credentials, click Review Information, and then click Import Storage Backend.

6. After the backends are added, the status changes to Available. These backends now have the information about the persistent volumes in the Tanzu Kubernetes cluster and the corresponding volumes on the ONTAP system.
7. For backup and restore across Tanzu Kubernetes clusters using Astra Control Center, you must provision an object storage bucket that supports the S3 protocol. Currently supported options are ONTAP S3, StorageGRID, AWS S3, and Microsoft Azure Blob storage. For the purpose of this installation, we are going to configure an AWS S3 bucket. Go to Buckets, click Add bucket, and select Generic S3. Enter the details about the S3 bucket and credentials to access it, click the checkbox Make this Bucket the Default Bucket for the Cloud, and then click Add.

**Choose the applications to protect**

After you have registered your Tanzu Kubernetes clusters, you can discover the applications that are deployed and manage them via the Astra Control Center.
Manage applications

1. After the Tanzu Kubernetes clusters and ONTAP backends are registered with the Astra Control Center, the control center automatically starts discovering the applications in all the namespaces that are using the storage class configured with the specified ONTAP backend.

2. Navigate to Apps > Discovered and click the dropdown menu next to the application you would like to manage using Astra. Then click Manage.

3. The application enters the Available state and can be viewed under the Managed tab in the Apps section.
Protect your applications

After application workloads are managed by Astra Control Center, you can configure the protection settings for those workloads.

Create an application snapshot

A snapshot of an application creates an ONTAP Snapshot copy and a copy of the application metadata that can be used to restore or clone the application to a specific point in time based on that Snapshot copy.

1. To take a snapshot of the application, navigate to the Apps > Managed tab and click the application you would like to make a Snapshot copy of. Click the dropdown menu next to the application name and click Snapshot.

2. Enter the snapshot details, click Next, and then click Snapshot. It takes about a minute to create the snapshot, and the status becomes Available after the snapshot is successfully created.
Create an application backup

A backup of an application captures the active state of the application and the configuration of it’s resources, converts them into files, and stores them in a remote object storage bucket.

1. For the backup and restore of managed applications in the Astra Control Center, you must configure superuser settings for the backing ONTAP systems as a prerequisite. To do so, enter the following commands.

   ONTAP::> export-policy rule modify -vserver ocp-trident -policyname default -ruleindex 1 -superuser sys
   ONTAP::> export-policy rule modify -policyname default -ruleindex 1 -anon 65534 -vserver ocp-trident

2. To create a backup of the managed application in the Astra Control Center, navigate to the Apps > Managed tab and click the application that you want to take a backup of. Click the dropdown menu next to the application name and click Backup.

   Enter the backup details, select the object storage bucket to hold the backup files, click Next, and, after reviewing the details, click Backup. Depending on the size of the application and data, the backup can take several minutes, and the status of the backup becomes Available after the backup is completed.
Restoring an application

At the push of a button, you can restore an application to the originating namespace in the same cluster or to a remote cluster for application protection and disaster recovery purposes.

1. To restore an application, navigate to the Apps > Managed tab and click the app in question. Click the dropdown menu next to the application name and click Restore.

2. Enter the name of the restore namespace, select the cluster you want to restore it to, and choose if you want to restore it from an existing snapshot or from a backup of the application. Click Next.
3. On the review pane, enter `restore` and click Restore after you have reviewed the details.

4. The new application goes to the Restoring state while Astra Control Center restores the application on the selected cluster. After all the resources of the application are installed and detected by Astra, the application goes to the Available state.
Cloning an application

You can clone an application to the originating cluster or to a remote cluster for dev/test or application protection and disaster recovery purposes. Cloning an application within the same cluster on the same storage backend uses NetApp FlexClone technology, which clones the PVCs instantly and saves storage space.

1. To clone an application, navigate to the Apps > Managed tab and click the app in question. Click the dropdown menu next to the application name and click Clone.

2. Enter the details of the new namespace, select the cluster you want to clone it to, and choose if you want to clone it from an existing snapshot, from a backup, or from the current state of the application. Click Next and then click Clone on the review pane after you have reviewed the details.
3. The new application goes to the Discovering state while Astra Control Center creates the application on the selected cluster. After all the resources of the application are installed and detected by Astra, the application goes to the Available state.

**Applications**

<table>
<thead>
<tr>
<th>Name</th>
<th>State</th>
<th>Protection</th>
<th>Cluster</th>
<th>Group</th>
<th>Discovered</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>magento-bef7f</td>
<td>Healthy</td>
<td>Unprotected</td>
<td>tigli-acc</td>
<td>magento-bef7f</td>
<td>2022/05/16 21:31 UTC</td>
<td></td>
</tr>
<tr>
<td>magento</td>
<td>Healthy</td>
<td>Partially protected</td>
<td>tigli-acc</td>
<td>magento</td>
<td>2022/05/09 18:20 UTC</td>
<td></td>
</tr>
</tbody>
</table>

**Astra Trident overview**

Unresolved directive in containers/vtwn_overview_trident.adoc - include://../include/containers_common_intro_sections.adoc[tags=trident_overview]

**Deploy Trident operator using Helm**

1. First set the location of the user cluster’s kubeconfig file as an environment variable so that you don’t have to reference it, because Trident has no option to pass this file.
2. Add the NetApp Astra Trident helm repository.

   [netapp-user@rhel7]$ helm repo add netapp-trident
   https://netapp.github.io/trident-helm-chart
   "netapp-trident" has been added to your repositories

3. Update the helm repositories.

   [netapp-user@rhel7]$ helm repo update
   Hang tight while we grab the latest from your chart repositories...
   ...Successfully got an update from the "netapp-trident" chart repository
   ...Successfully got an update from the "bitnami" chart repository
   Update Complete. ⬛Happy Helming!⬜

4. Create a new namespace for the installation of Trident.

   [netapp-user@rhel7]$ kubectl create ns trident

5. Create a secret with DockerHub credentials to download the Astra Trident images.

   [netapp-user@rhel7]$ kubectl create secret docker-registry docker-registry-cred
   --docker-server=docker.io --docker-username=netapp-solutions-tme --docker-password=xxxxxx -n trident

6. For user or workload clusters managed by TKGS (vSphere with Tanzu) or TKG with management cluster deployments, complete the following procedure to install Astra Trident:

   a. Ensure that the logged in user has the permissions to create service accounts in trident namespace
      and that the service accounts in trident namespace have the permissions to create pods.

   b. Run the below helm command to install Trident operator in the namespace created.

      [netapp-user@rhel7]$ helm install trident netapp-trident/trident-operator
      -n trident --set imagePullSecrets[0]=docker-registry-cred

7. For a user or workload cluster managed by TKGI deployments, run the following helm command to install Trident operator in the namespace created.
8. Verify that the Trident pods are up and running.

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>trident-csi-6vv62</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
</tr>
<tr>
<td>14m</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trident-csi-cfd844bcc-sqhcg</td>
<td>6/6</td>
<td>Running</td>
<td>0</td>
</tr>
<tr>
<td>12m</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trident-csi-dfcmz</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
</tr>
<tr>
<td>14m</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trident-csi-pb2n7</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
</tr>
<tr>
<td>14m</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trident-csi-qsw6z</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
</tr>
<tr>
<td>14m</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trident-operator-67c94c4768-xw978</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
</tr>
<tr>
<td>14m</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
[netapp-user@rhel7]$ ./tridentctl -n trident version
+----------------+----------------+----------------+----------------+
| SERVER VERSION | CLIENT VERSION |
+----------------+----------------+
| 22.04.0        | 22.04.0        |
+----------------+----------------+
```

Create storage-system backends

After completing the Astra Trident Operator install, you must configure the backend for the specific NetApp storage platform you are using. Follow the links below to continue the setup and configuration of Astra Trident.

- NetApp ONTAP NFS
- NetApp ONTAP iSCSI

NetApp ONTAP NFS configuration

To enable Trident integration with the NetApp ONTAP storage system via NFS, you must create a backend that enables communication with the storage system. We configure a basic backend in this solution, but if you are looking for more customized options, visit the documentation [here](#).

Create an SVM in ONTAP

1. Log into ONTAP System Manager, navigate to Storage > Storage VMs, and click Add.
2. Enter a name for the SVM, enable the NFS protocol, check the Allow NFS Client Access checkbox, and add the subnets that your worker nodes are on in the export policy rules for allowing the volumes to be mounted as PVs in your workload clusters.

If you are using NAT’ed deployment of user clusters or workload clusters with NSX-T, you need to add the Egress subnet (in the case of TKGS0 or the Floating IP subnet (in the case of TKGI) to the export policy rules.

3. Provide the details for data LIFs and the details for SVM administration account, and then click Save.
4. Assign the aggregates to an SVM. Navigate to Storage > Storage VMs, click the ellipsis next to the newly created SVM and then click Edit. Check the Limit Volume Creation to Preferred Local Tiers checkbox and attach the required aggregates to it.
5. In case of NAT’ed deployments of user or workload clusters on which Trident is to be installed, the storage mount request might arrive from a non-standard port due to SNAT. By default, ONTAP only allows the volume mount requests when originated from root port. Thus, log into ONTAP CLI and modify the setting to
allow mount requests from non-standard ports.

```
ontap-01> vserver nfs modify -vserver tanzu_svm -mount-rootonly disabled
```

Create backends and StorageClasses

1. For NetApp ONTAP systems serving NFS, create a backend config file on the jumphost with the backendName, managementLIF, dataLIF, svm, username, password, and other details.

```
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "ontap-nas+10.61.181.221",
    "managementLIF": "172.21.224.201",
    "dataLIF": "10.61.181.221",
    "svm": "trident_svm",
    "username": "admin",
    "password": "password"
}
```

It is a best practice to define the custom backendName value as a combination of the storageDriverName and the dataLIF that is serving NFS for easy identification.

2. Create the Trident backend by running the following command.

```
[netapp-user@rhel7]$ ./tridentctl -n trident create backend -f backend-ontap-nas.json
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>STORAGE DRIVER</th>
<th>UUID</th>
</tr>
</thead>
<tbody>
<tr>
<td>ontap-nas+10.61.181.221</td>
<td>ontap-nas</td>
<td>be7a619d-c81d-445c-b80c-5c87a73c5b1e</td>
</tr>
<tr>
<td>online</td>
<td>0</td>
<td>136509020512</td>
</tr>
</tbody>
</table>

3. With the backend created, you must next create a storage class. The following sample storage class definition highlights the required and basic fields. The parameter `backendType` should reflect the storage driver from the newly created Trident backend.
4. Create the storage class by running the kubectl command.

```bash
[netapp-user@rhel7 trident-installer]$ kubectl create -f storage-class-nfs.yaml
storageclass.storage.k8s.io/ontap-nfs created
```

5. With the storage class created, you must then create the first persistent volume claim (PVC). A sample PVC definition is given below. Make sure that the `storageClassName` field matches the name of the storage class just created. The PVC definition can be further customized as required depending upon the workload to be provisioned.

```yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-nfs
```

6. Create the PVC by issuing the kubectl command. Creation can take some time depending on the size of the backing volume being created, so you can watch the process as it completes.

```bash
[netapp-user@rhel7 trident-installer]$ kubectl create -f pvc-basic.yaml
persistentvolumeclaim/basic created
```

```
[netapp-user@rhel7 trident-installer]$ kubectl get pvc
NAME     STATUS   VOLUME                                      CAPACITY
ACCESS MODES STORAGECLASS AGE
basic     Bound   pvc-b4370d37-0fa4-4c17-bd86-94f96c94b42d 1Gi
RWO       ontap-nfs 7s
```
NetApp ONTAP iSCSI configuration

To integrate NetApp ONTAP storage system with VMware Tanzu Kubernetes clusters for persistent volumes via iSCSI, the first step is to prepare the nodes by logging into each node and configuring the iSCSI utilities or packages to mount iSCSI volumes. To do so, follow the procedure laid out in this link.

NetApp does not recommend this procedure for NAT’ed deployments of VMware Tanzu Kubernetes clusters.

TKGI uses Bosh VMs as nodes for Tanzu Kubernetes clusters that run immutable configuration images, and any manual changes of iSCSI packages on Bosh VMs do not remain persistent across reboots. Therefore, NetApp recommends using NFS volumes for persistent storage for Tanzu Kubernetes clusters deployed and operated by TKGI.

After the cluster nodes are prepared for iSCSI volumes, you must create a backend that enables communication with the storage system. We configured a basic backend in this solution, but, if you are looking for more customized options, visit the documentation here.

Create an SVM in ONTAP

To create an SVM in ONTAP, complete the following steps:

1. Log into ONTAP System Manager, navigate to Storage > Storage VMs, and click Add.
2. Enter a name for the SVM, enable the iSCSI protocol, and then provide details for the data LIFs.
3. Enter the details for the SVM administration account, and then click Save.

### Access Protocol

**SMB/CIFS, NFS, S3**

- **iSCSI**

- **Enable iSCSI**

### NETWORK INTERFACE

#### K8s-Ontap-01

<table>
<thead>
<tr>
<th>IP ADDRESS</th>
<th>SUBNET MASK</th>
<th>GATEWAY</th>
<th>BROADCAST DOMAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.61.181.231</td>
<td>24</td>
<td>10.61.181.1</td>
<td>Default</td>
</tr>
</tbody>
</table>

- Use the same subnet mask, gateway, and broadcast domain for all of the following interfaces

<table>
<thead>
<tr>
<th>IP ADDRESS</th>
<th>SUBNET MASK</th>
<th>GATEWAY</th>
<th>BROADCAST DOMAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.61.181.232</td>
<td>24</td>
<td>10.61.181.1</td>
<td>Default</td>
</tr>
</tbody>
</table>
4. To assign the aggregates to the SVM, navigate to Storage > Storage VMs, click the ellipsis next to the newly created SVM, and then click Edit. Check the Limit Volume Creation to Preferred Local Tiers checkbox, and attach the required aggregates to it.
Create backends and StorageClasses

1. For NetApp ONTAP systems serving NFS, create a backend config file on the jumphost with the backendName, managementLIF, dataLIF, svm, username, password, and other details.
2. Create the Trident backend by running the following command.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create backend -f backend-ontap-san.json
```

|          NAME          | STORAGE DRIVER |                 UUID                     |
| STATE  | VOLUMES |                                                  |
|------------------------+-------------+------------------------------------------|
| ontap-san+10.61.181.231 | ontap-san  | 6788533c-7fea-4a35-b797-fb9bb3322b91 | online | 0 |

3. After you create a backend, you must next create a storage class. The following sample storage class definition highlights the required and basic fields. The parameter `backendType` should reflect the storage driver from the newly created Trident backend. Also note the name-field value, which must be referenced in a later step.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-iscsi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
```

There is an optional field called `fsType` that is defined in this file. In iSCSI backends, this value can be set to a specific Linux filesystem type (XFS, ext4, and so on) or can be deleted to allow Tanzu Kubernetes clusters to decide what filesystem to use.
4. Create the storage class by running the `kubectl` command.

```
[netapp-user@rhel7 trident-installer]$ kubectl create -f storage-class-iscsi.yaml
storageclass.storage.k8s.io/ontap-iscsi created
```

5. With the storage class created, you must then create the first persistent volume claim (PVC). A sample PVC definition is given below. Make sure that the `storageClassName` field matches the name of the storage class just created. The PVC definition can be further customized as required depending upon the workload to be provisioned.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-iscsi
```

6. Create the PVC by issuing the `kubectl` command. Creation can take some time depending on the size of the backing volume being created, so you can watch the process as it completes.

```
[netapp-user@rhel7 trident-installer]$ kubectl create -f pvc-basic.yaml
persistentvolumeclaim/basic created
[netapp-user@rhel7 trident-installer]$ kubectl get pvc
NAME    STATUS   VOLUME                                     CAPACITY
ACCESS M  STORAGECLASS   AGE
basic    Bound    pvc-7ceaclba-0189-43c7-8f98-094719f7956c   1Gi
RWO      ontap-iscsi      3s
```

**Videos and demos: VMware Tanzu with NetApp**

The following videos demonstrate some of the capabilities described in this document:

- Use Astra Trident to Provision Persistent Storage in VMware Tanzu - VMware Tanzu with NetApp
- Use Astra Control Center to Clone Applications in VMWare Tanzu - VMware Tanzu with NetApp
These demos were recorded as a tech preview using version 1.3.1 of TKG and version 21.12 of Astra Control Center. Please see the Support Matrix for official supported versions.

Additional Information: VMware Tanzu with NetApp

To learn more about the information described in this document, review the following websites:

- NetApp Documentation
  https://docs.netapp.com/
- Astra Trident Documentation
  https://docs.netapp.com/us-en/trident/
- NetApp Astra Control Center Documentation
  https://docs.netapp.com/us-en/astra-control-center/
- Ansible Documentation
  https://docs.ansible.com/
- VMware Tanzu Documentation
- VMware Tanzu Kubernetes Grid Documentation
- VMware Tanzu Kubernetes Grid Service Documentation
  https://docs.vmware.com/en/VMware-vSphere/7.0/vmware-vsphere-with-tanzu/GUID-152BE7D2-E227-4DAA-B527-557B564D9718.html
- VMware Tanzu Kubernetes Grid Integrated Edition Documentation

Archived Solutions

Anthos on bare metal
Data Migration and Data Protection

Data Migration

Best-Practice Guidelines for NetApp XCP

TR-4863: Best-Practice Guidelines for NetApp XCP - Data Mover, File Migration, and Analytics

Karthikeyan Nagalingam, NetApp

This document provides NetApp XCP best-practice guidelines and a test scenario-based solution. These best practices cover the migration workflow for on-premises as well as cloud, file-system analytics, troubleshooting, and performance tuning of XCP. The test-scenario section covers customer use cases and their requirements, the NetApp solution using XCP, and benefits to the customer.

NetApp XCP

NetApp XCP transfers data by using multithreads and customizable features. It is designed for three major use cases: data move or migration, file-system analytics, and fast directory tree deletion.

Data move or migration

NetApp XCP transfers data from any NAS to NetApp NAS. This process consists of four major operations: scan, copy, sync, and verify. There are some additional features that help the data monitoring and transfer:

- **Scan.** Provides a high-level layout of NAS and MapR/HDFS data.
- **Copy.** Performs a baseline data transfer.
- **Sync.** Performs the incremental data transfer.
- **Verify.** Performs a thorough verification of the target.
- **Show (optional).** Discovers NAS shares.

The following figure illustrates XCP data migration and replication operations.
File-system analytics

NetApp XCP natively enables you to identify, scrutinize, and analyze unstructured data to improve insights—a key requirement for enterprise customers who want to use those insights for better planning, to operationalize high-value digital assets, and for data governance through reporting and assessment.

Customers that deal with sensitive data can use NetApp XCP to answer typical operational questions, such as the following:

- Where is my data?
- How much data and what types of files do we have?
- What data is actively used and how much is dormant?

The following figure illustrates NetApp XCP file analytics communication from the GUI.

Delete

It can be very challenging for storage teams and Electronic Design Automation (EDA) workloads to clean up large directories, whether it’s stale data or test data that needs to be cleaned to recover storage space. XCP provides a fast delete functionality that can delete a complete directory tree. The NetApp XCP Delete function removes files and folders from a given NAS path. You can leverage the match filters to delete a specific set of files and folders. For a large number of files and folders, you can use the Force option, which does not require
a confirmation to delete.

**Live Source Migration support**

Live Source Migration support included in XCP 1.7 allows migration from a data source that is in active use (read and write activity). XCP leaves out files that are being used during the migration job, such as copy and sync running, and skipped files information is captured in the XCP log.

This feature supports changes on the source but does not support changes on the destination. During migration, the destination should not be active. Live Source Migration support is only available for NFS migrations.

![i] No special settings are required for Live Source Migrations.

**Prerequisites for XCP**

Before you deploy NetApp XCP, the following prerequisites must be met:

1. Verify the NFS ports used by the NFS server by running the following command:

   ```
   rpcinfo -p < NAS IP or on-prem nfs data LIF ip >
   ```

2. To access the location where you execute the XCP operations, such as on-premises or cloud instances (for example, Azure, AWS, or Google virtual machine [VM] instances), open the firewall ports for the NFS ports.

3. Verify that the NFS port is accessible from the XCP server by using the telnet command `<on-prem nfs data LIF ip or NAS IP > 2049`. The default port is 2049. If your environment has a different port, use that IP.

4. For NFS, verify that the shares are accessible from the XCP server by using the `showmount -e < NAS IP >` command.

5. Increase the number of inodes on the destination volume to more than the file count (number of files) on the source files.

6. Download the XCP license from the [NetApp XCP License Portal](#).
   a. You must have a NetApp account in mysupport.netapp.com or you can register for free.
   b. Download the license and have it ready.

7. Create one NFS share on-premises for each Azure NetApp volume or for the Cloud Volume Service (premium service level) in cloud for the XCP catalog.

8. Create an NAS volume and configure the share for the data destination.

9. For multiple XCP instances, you must have one or more servers or cloud instances to transfer the data from multiple source folders or files to the destination.

10. The maxdir size (default is 308MB) defines the maximum file count (approximately one million) in a single folder. Increase the maxdir size value to increase the file count. Increasing the value has an effect on additional CPU cycles.

11. In the cloud, NetApp recommends that you have ExpressRoute (Azure), Direct Connect (AWS), or Cloud Interconnect (GCP) between on-premises and cloud.
Migration workflow

Migration has different phases to follow for better planning and completion of the migration. To migrate data from third-party NAS storage or directly attached NAS exported storage using NetApp XCP, follow the migration guidelines provided in this section.

The following figure illustrates the migration workflow from any NAS to NetApp NAS.

On-premises

The migration workflow from any NAS to NetApp NAS includes the following steps:

1. Discover the NAS shares and data.
2. Scan the data and produce a report to find the layout of the data.
3. Create a baseline by running the XCP Copy command. For faster migrations, select more XCP instances and split the workload at the subfolder level to initiate parallel migration jobs.
4. For incremental updates, use XCP sync until the change rate is low for the cutover window.
5. Mark the source as read-only to perform a final sync by running the XCP sync command to complete the migration.
6. To verify that the data transferred correctly, compare the source and destination by running the `xcp verify` command.

Cloud

For the cloud, you can follow a similar on-premises migration workflow if the connectivity between on-premises and the cloud is direct connect (AWS), ExpressRoute (Azure), or cloud interconnect (GCP).

The following figure illustrates the migration workflow from on-premises to the cloud.
If there is no direct internet connection between on-premises and the cloud, you must transfer the data from on-premises to the cloud through an offline data transport method such as truck. Each cloud service provider has a different method with different terminology to move data to their data center.

The following figure depicts the data mover solution for on-premises to Azure without ExpressRoute.

You can use a similar architecture with the respective components from the various cloud service providers.
File analytics

The NetApp XCP file analytics GUI helps to run file system scans by using XCP at the back end and visualizing statistics such as graphs and views for any NAS (NFS, SMB) file system. Starting in 1.6, XCP can be run as a service with the help of simple deployment steps by using the Configure and systemctl options. The XCP Configure option guides you to install and configure Postgres and a web server as well as collect credentials. The systemctl option runs XCP as a service for REST API communications from the GUI.

The following figure illustrates the XCP file analytics flow.

For more information about the high-level architecture of XCP file analytics, GUI-based dashboard views such as stats view, and file distribution view details, see the blog post NetApp XCP 1.6 Delivers Open File Analytics and Infrastructure Improvements.

There is a limited GUI in XCP 1.6 for customized graphs. To create the required graphs, you can use the CLI to run the \texttt{xcp scan} command with matching filters. See the following examples.

1. Generate a list of files modified beyond a year by using \texttt{xcp scan} and the \texttt{-match} filter with the space consumed.
2. Find the space used by files that are more than one year old.

[root@ch-vm-cent7-2 linux]# ./xcp -du -match "modified > 1*year"
xcp: WARNING: CPU count is only 1!

52.5KiB
data_for_analysis/benchmarks/benchmarks/Macro_Scope_1_benchmark.out
28.5KiB
data_for_analysis/benchmarks/benchmarks/CollectedGroup_6_benchmark.out
28.5KiB data_for_analysis/benchmarks/benchmarks/Foreach_11_benchmark.out
153KiB
data_for_analysis/benchmarks/benchmarks/SecondarySort_9_benchmark.out
412KiB
data_for_analysis/benchmarks/benchmarks/CoGroupFlatten_6_benchmark.out
652KiB data_for_analysis/benchmarks/benchmarks/Iterator_1_benchmark.out
652KiB
data_for_analysis/benchmarks/benchmarks/LoaderDefaultDir_1_benchmark.out
652KiB data_for_analysis/benchmarks/benchmarks/Order_4_benchmark.out
28.5KiB
data_for_analysis/benchmarks/benchmarks/MapPartialAgg_4_benchmark.out/2
28.5KiB
data_for_analysis/benchmarks/benchmarks/CastScalar_11_benchmark.out/2
1.29MiB data_for_analysis/benchmarks/benchmarks/Order_18_benchmark.out
652KiB
data_for_analysis/benchmarks/benchmarks/FilterBoolean_5_benchmark.out
20.5KiB
data_for_analysis/benchmarks/benchmarks/Macro_DefinitionAndInline_5_benchmark.out/2
628KiB data_for_analysis/benchmarks/benchmarks/Types_29_benchmark.out
...
< console output removed due to page space size >
...
3.18MiB data_for_analysis/benchmarks/benchmarks/hadoop10
340KiB data_for_analysis/benchmarks/benchmarks/Split_5_benchmark.out
5.90GiB data_for_analysis/benchmarks/benchmarks
6.56GiB data_for_analysis/benchmarks
6.56GiB data_for_analysis

Filtered: 488 did not match

Xcp command : xcp -du -match modified > 1*year
192.168.89.110:/ifs/data_for_analysis/
Stats : 5,055 scanned, 4,567 matched
Speed : 1.10 MiB in (1.36 MiB/s), 110 KiB out (135 KiB/s)
Total Time : 0s.
STATUS : PASSED
[root@ch-vm-cent7-2 linux]#
3. Find the total size and graphical view of data that was modified more than one year ago.

```
[root@ch-vm-cent7-2 linux]# ./xcp -stats -match "modified > 1*year" -html 192.168.89.110:/ifs/data_for_analysis/ > modified_morethan_year_stats.html
XCP 1.6.1; (c) 2020 NetApp, Inc.; Licensed to Karthikeyan Nagalingam

xcp: WARNING: CPU count is only 1!

Xcp command : xcp -stats -match modified > 1*year -html 192.168.89.110:/ifs/data_for_analysis/
Stats       : 5,055 scanned, 4,567 matched
Speed       : 1.10 MiB in (919 KiB/s), 110 KiB out (89.1 KiB/s)
Total Time  : 1s.
STATUS      : PASSED
[root@ch-vm-cent7-2 linux]#
```

The following report is a custom example scan of files that were modified more than one year ago.
Deployment steps

This section covers deployment steps for NetApp XCP for data transfer.
Test bed details

The following table provides the details of the test bed that was used for this deployment and performance validation.

<table>
<thead>
<tr>
<th>Solution components</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>XCP version 1.7</td>
<td>• One Linux server - Linux (RHEL 7.9 or RHEL 8)</td>
</tr>
<tr>
<td></td>
<td>• One Windows server – Windows Server 2019 standard</td>
</tr>
<tr>
<td>NetApp AFF storage array HA pair for the source volume</td>
<td>• AFF8080</td>
</tr>
<tr>
<td></td>
<td>• NetApp ONTAP 9</td>
</tr>
<tr>
<td></td>
<td>• NFS protocol</td>
</tr>
<tr>
<td>NetApp AFF storage array HA pair for destination volume</td>
<td>• AFF A800</td>
</tr>
<tr>
<td></td>
<td>• ONTAP 9</td>
</tr>
<tr>
<td></td>
<td>• NFS protocol</td>
</tr>
<tr>
<td>Fujitsu PRIMERGY RX2540 server</td>
<td>Each equipped with:</td>
</tr>
<tr>
<td></td>
<td>* 48 CPUs</td>
</tr>
<tr>
<td></td>
<td>* Intel Xeon</td>
</tr>
<tr>
<td></td>
<td>* 256GB physical memory</td>
</tr>
<tr>
<td></td>
<td>* 10GbE dual port</td>
</tr>
<tr>
<td>Networking</td>
<td>10GbE</td>
</tr>
</tbody>
</table>

Deployment steps - NAS

To deploy NetApp XCP for data transfer, first install and activate the XCP software on the destination location. You can review the details in the NetApp XCP User Guide. To do so, complete the following steps:

1. Meet the prerequisites as detailed in the section “Prerequisites for XCP.”
2. Download the XCP software from the NetApp XCP (Downloads) page.
3. Copy the downloaded XCP tar files to the XCP server.

```
# scp Documents/OneDrive\ -\ NetApp\ 
Inc/XCP/software/1.6.1/NETAPP_XCP_1.6.1.tgz 
mailto:root@10.63.150.53:/usr/src
```

4. Untar the tarfile.

```
[root@mastr-53 src]# tar -zxvf NETAPP_XCP_1.6.1.tgz
```

5. Download the license from https://xcp.netapp.com/license/xcp.xwic and copy to the XCP server.
6. Activate the license.
7. Find the source NFS port and destination NFS server. The default port is 2049.

```
[root@mastr-53 ~]# rpcinfo -p 10.63.150.213
[root@mastr-53 ~]# rpcinfo -p 10.63.150.63
```

8. Check the NFS connection. Check the NFS server (for both source and destination) by using telnet to the NFS server port.

```
[root@mastr-53 ~]# telnet 10.63.150.127 2049
[root@mastr-53 ~]# telnet 10.63.150.63 2049
```

9. Configure the catalog.
   
a. Create an NFS volume and export NFS for the XCP catalog. You can also leverage the operating system NFS export for XCP catalog.

```
A800-Node1-2::> volume create -vserver Hadoop_SVM -volume xcpcatalog
-aggregate aggr_Hadoop_1 -size 50GB -state online -junction-path
/xcpcatalog -policy default -unix-permissions ---rwxr-xr-x -type RW
-snapshot-policy default -foreground true
A800-Node1-2::> volume mount -vserver Hadoop_SVM -volume
xcpcatalog_vol -junction-path /xcpcatalog
```

b. Check the NFS export.

```
[root@mastr-53 ~]# showmount -e 10.63.150.63 | grep xcpcata
/xcpcatalog (everyone)
```

c. Update xcp.ini.

```
[root@mastr-53 ~]# cat /opt/NetApp/xFiles/xcp/xcp.ini
# Sample xcp config
[xcp]
catalog = 10.63.150.64:/xcpcatalog
```

10. Find the source NAS exports by using `xcp show`. Look for:

```
== NFS Exports ==
== Attributes of NFS Exports ==
```

```
[root@mastr-53 linux]# ./xcp show 10.63.150.127
== NFS Exports ==
<check here>
== Attributes of NFS Exports ==
<check here>
```

11. (Optional) Scan the source NAS data.

```
[root@mastr-53 linux]# ./xcp scan -newid xcpscantest4 -stats
10.63.150.127:/xcpsrc_vol
```

Scanning the source NAS data helps you understand the data layout and find any potential issues for migration. The XCP scanning operation time is proportional to the number of files and the directory depth. You can skip this step if you are familiar with your NAS data.

12. Check the report created by `xcp scan`. Search mainly for unreadable folders and unreadable files.

```
[root@mastr-53 linux]# mount 10.63.150.64:/xcpcatalog /xcpcatalog
(base) nkarthik-mac-0:~ karthikeyannagalingam$ scp -r root@10.63.150.53:/xcpcatalog/catalog/indexes/xcpscantest4 Documents/OneDrive\ -\ NetApp\ Inc/XCP/customers/reports/
```

13. (Optional) Change the inode. View the number of inodes and modify the number based on the number of files to migrate or copy for both catalog and destination volumes (if required).

```
A800-Node1-2::> volume show -volume xcpcatalog -fields files,files-used
A800-Node1-2::> volume show -volume xcpdest -fields files,files-used
A800-Node1-2::> volume modify -volume xcpcatalog -vserver A800-Node1_vs1 -files 2000000
Volume modify successful on volume xcpcatalog of Vserver A800-Node1_vs1.
A800-Node1-2::> volume show -volume xcpcatalog -fields files,files-used
```

14. Scan the destination volume.

```
[root@mastr-53 linux]# ./xcp scan -stats 10.63.150.63:/xcpdest
```
15. Check the source and destination volume space.

```
[root@mastr-53 ~]# df -h /xcpsrc_vol
[root@mastr-53 ~]# df -h /xcpdest/
```

16. Copy the data from source to destination by using `xcp copy` and check the summary.

```
[root@mastr-53 linux]# ./xcp copy -newid create_Sep091599198212
10.63.150.127:/xcpsrc_vol 10.63.150.63:/xcpdest
<command inprogress results removed>
Xcp command : xcp copy -newid create_Sep091599198212 -parallel 23
10.63.150.127:/xcpsrc_vol 10.63.150.63:/xcpdest
Stats       : 9.07M scanned, 9.07M copied, 118 linked, 9.07M indexed, 173 giants
Speed       : 1.57 TiB in (412 MiB/s), 1.50 TiB out (392 MiB/s)
Total Time  : 1h6m.
STATUS      : PASSED
[root@mastr-53 linux]#
```

- By default, XCP creates seven parallel processes to copy the data. This can be tuned.

- NetApp recommends that the source volume be read only. In real time, the source volume is a live, active file system. The `xcp copy` operation might fail because NetApp XCP does not support a live source that is continuously changed by an application.

For Linux, XCP requires an Index ID because XCP Linux performs cataloging.

17. (Optional) Check the inodes on the destination NetApp volume.

```
A800-Node1-2::> volume show -volume xcpdest -fields files,files-used

vserver        volume  files    files-used
-------------- ------- -------- ----------
A800-Node1_vs1 xcpdest 21251126 15039685
A800-Node1-2::>
```

18. Perform the incremental update by using `xcp sync`. 

For this document, to simulate real-time, the one million files in the source data were renamed, and then the updated files were copied to the destination by using xcp sync. For Windows, XCP needs both source and destination paths.

19. Validate data transfer. You can validate that the source and destination have the same data by using xcp verify.

XCP documentation provides multiple options (with examples) for the scan, copy, sync, and verify operations. For more information, see the NetApp XCP User Guide.

Windows customers should copy the data by using access control lists (ACLs). NetApp recommends using the command xcp copy -acl -fallbackuser\<username> -fallbackgroup\<username or groupname> <source> <destination>. To maximum performance, considering the source volume that has SMB data with ACL and the data accessible by both NFS and SMB, the target must be an NTFS volume. Using XCP (NFS version), copy the data from the Linux server and execute the XCP (SMB version) sync with the -acl and -nodata options from the Windows server to copy the ACLs from source data to the target SMB data.

For detailed steps, see Configuring 'Manage Auditing and Security Log' Policy.

Deployment steps - HDFS/MapRFS data migration

In this section, we discuss the new XCP feature called Hadoop Filesystem Data Transfer to NAS, which migrates data from HDFS/MapRFS to NFS and vice versa.

Prerequisites

For the MapRFS/HDFS feature, you must perform the following procedure in a non-root user environment. Normally the non-root user is hdfs, mapr, or a user who has permission to make changes in the HDFS and MapRFS filesystem.
1. Set the CLASSPATH, HADOOP_HOME, NHDFS_LIBJVM_PATH, LB_LIBRARY_PATH, and NHDFS_LIBHDFS_PATH variables in the CLI or the .bashrc file of the user along with the xcp command.
   - NHDFS_LIBHDFS_PATH points to the libhdfs.so file. This file provides HDFS APIs to interact and manipulate the HDFS/MapRFS files and filesystem as a part of the Hadoop distribution.
   - NHDFS_LIBJVM_PATH points to the libjvm.so file. This is a shared JAVA virtual machine library in the jre location.
   - CLASSPATH points to all jars files using (Hadoop classpath --glob) values.
   - LD_LIBRARY_PATH points to the Hadoop native library folder location.

See the following sample based on a Cloudera cluster.

```bash
export CLASSPATH=$(hadoop classpath --glob)
export LD_LIBRARY_PATH=/usr/java/jdk1.8.0_181-cloudera/jre/lib/amd64/server/
export HADOOP_HOME=/opt/cloudera/parcels/CDH-6.3.4-1.cdh6.3.4.p0.6751098/
#export HADOOP_HOME=/opt/cloudera/parcels/CDH/
export NHDFS_LIBJVM_PATH=/usr/java/jdk1.8.0_181-cloudera/jre/lib/amd64/server/libjvm.so
export NHDFS_LIBHDFS_PATH=$HADOOP_HOME/lib64/libhdfs.so
```

In this release, we support XCP scan, copy, and verify operations and data migration from HDFS to NFS. You can transfer data from a data lake cluster single worker node and multiple worker nodes. In the 1.8 release, root and non-root users can perform data migration.

**Deployment steps - Non-root user migrates HDFS/MaprFS data to NetApp NFS**

1. Follow the same steps mentioned from 1-9 steps from steps for deployment section.
2. In the following example, the user migrates data from HDFS to NFS.
   a. Create a folder and files (using hadoop fs -copyFromLocal) in HDFS.
b. Check permissions in the HDFS folder.

```
[root@n138 ~]# su - tester -c 'hadoop fs -ls hdfs:///tmp/testerfolder_src/util-linux-2.23.2/mohankarthikhdfs_src'
Found 2 items
-rw-r--r--   3 tester supergroup         14 2021-11-16 17:01 hdfs:///tmp/testerfolder_src/util-linux-2.23.2/mohankarthikhdfs_src/a_hdfs.txt
-rw-r--r--   3 tester supergroup         16 2021-11-16 17:01 hdfs:///tmp/testerfolder_src/util-linux-2.23.2/mohankarthikhdfs_src/b_hdfs.txt
```

c. Create a folder in NFS and check permissions.

```
[root@n138 ~]# su - tester -c 'hadoop fs -mkdir /tmp/testerfolder_src/util-linux-2.23.2/mohankarthikhdfs_src'
[root@n138 ~]# su - tester -c 'hadoop fs -ls -d /tmp/testerfolder_src/util-linux-2.23.2/mohankarthikhdfs_src'
drwxr-xr-x   - tester supergroup          0 2021-11-16 16:52 /tmp/testerfolder_src/util-linux-2.23.2/mohankarthikhdfs_src
[root@n138 ~]# su - tester -c "echo 'testfile hdfs' > /tmp/a_hdfs.txt"
[root@n138 ~]# su - tester -c "echo 'testfile hdfs 2' > /tmp/b_hdfs.txt"
[root@n138 ~]# ls -ltrah /tmp/*_hdfs.txt
-rw-rw-r-- 1 tester tester 14 Nov 16 17:00 /tmp/a_hdfs.txt
-rw-rw-r-- 1 tester tester 16 Nov 16 17:00 /tmp/b_hdfs.txt
[root@n138 ~]# su - tester -c 'hadoop fs -copyFromLocal /tmp/*_hdfs.txt hdfs:///tmp/testerfolder_src/util-linux-2.23.2/mohankarthikhdfs_src'
[root@n138 ~]#
```
d. Copy the files from HDFS to NFS using XCP, and check permissions.

```
[root@n138 ~]# su - tester -c 'mkdir /xcpsrc_vol/mohankarthiknfs_dest'
[root@n138 ~]# su - tester -c 'ls -l /xcpsrc_vol/mohankarthiknfs_dest'
total 0
[root@n138 ~]# su - tester -c 'ls -d /xcpsrc_vol/mohankarthiknfs_dest'
/xcpsrc_vol/mohankarthiknfs_dest
[root@n138 ~]# su - tester -c 'ls -ld /xcpsrc_vol/mohankarthiknfs_dest'
drwxr-xr-x 2 tester tester 4096 Nov 16 14:32 /xcpsrc_vol/mohankarthiknfs_dest
[root@n138 ~]#
```

```
Copy the files from HDFS to NFS using XCP, and check permissions.

```
[root@n138 ~]# su - tester -c '/usr/src/hdfs_nightly/xcp/linux/xcp copy -chown hdfs:///tmp/testerfolder_src/util-linux-2.23.2/mohankarthikhdfs_src/10.63.150.126:/xcpsrc_vol/mohankarthiknfs_dest'
XCP Nightly_dev; (c) 2021 NetApp, Inc.; Licensed to Karthikeyan Nagalingam [NetApp Inc] until Wed Feb 9 13:38:12 2022
xcp: WARNING: No index name has been specified, creating one with name: autoname_copy_2021-11-16_17.04.03.652673
Xcp command : xcp copy -chown hdfs:///tmp/testerfolder_src/util-linux-2.23.2/mohankarthikhdfs_src/10.63.150.126:/xcpsrc_vol/mohankarthiknfs_dest
Stats : 3 scanned, 2 copied, 3 indexed
Speed : 3.44 KiB in (650/s), 80.2 KiB out (14.8 KiB/s)
Total Time : 5s.
STATUS : PASSED
[root@n138 ~]# su - tester -c 'ls -l /xcpsrc_vol/mohankarthiknfs_dest'
total 0
-rw-r--r-- 1 tester supergroup 14 Nov 16 17:01 a_hdfs.txt
-rw-r--r-- 1 tester supergroup 16 Nov 16 17:01 b_hdfs.txt
[root@n138 ~]# su - tester -c 'ls -ld /xcpsrc_vol/mohankarthiknfs_dest'
drwxr-xr-x 2 tester tester 4096 Nov 16 14:32 /xcpsrc_vol/mohankarthiknfs_dest
[root@n138 ~]#
```
Sizing guidelines

This section provides the approximate time to perform the XCP copy and XCP sync operations with a different file size of one million files for NFS.

Time estimate based on testing

The tests for the XCP copy and sync operations used the same test bed that was used for deployment. One million files of three sets of 8K, 16K, and 1MB files were created and the changes were performed in real time. The XCP sync function performed the differential incremental updates from the source to the target at the file level. The incremental update operation is one or more of these four operations: rename existing files and folders, append data to existing files, delete files and folders, and include additional hard, soft, and multilinks. For test purposes, we focused on the rename, append, delete, and links operations. In other words, the modification operations such as rename, append, and delete were performed at a change rate of 10% to 90% on one million files.

The following figure shows the results of the XCP copy operation.

The following figure shows the results of the XCP Sync rename and link operations.

The file size is not propositional to the XCP sync completion time for transferring the renamed source files; the graphs are linear.
The link types are soft links, hard links, and multi-links. Soft links are considered normal files. The size of the files is not relevant for the time to complete the XCP sync operation.

The following figures show the results of the XCP sync append and delete operations.

For the append and delete operations, large file sizes take more time compared to small file sizes. The time to complete the operation is linear to the percentage of append and delete changes.

**Comparing XCP 1.6.1 to XCP 1.5**

Compared to previous versions, XCP 1.6.3 and 1.7 provides improved performance. The following section shows a sync performance comparison between XCP 1.6.3 and 1.7 for 8K, 16K, and 1MB sizes of one million files.

The following figures show the results of the XCP sync performance for XCP 1.6.3 versus 1.7 (with an 8K size of one million files).
The following figure shows the results of the XCP sync performance for XCP 1.6.1 versus 1.5 (with a 16K size of one million files).
The following figure shows the results of the XCP sync performance for XCP 1.6.1 versus 1.5 with a 1MB size of one million files.
On average, the XCP 1.7 performance improved on or was similar to XCP 1.6.3 for the `xcp sync` differential incremental update–rename, append, link, and delete operations with a 1MB size of one million files.

Based on this performance validation, NetApp recommends using XCP 1.7 for your data migration on-premises and in the cloud.

**Performance tuning**

This section provides some of the tuning parameters that help to improve the performance of XCP operations:

- For better scaling and to distribute the workload across multiple XCP instances, split the subfolders for each XCP instance for the migration and data transfer.
- XCP can use maximum CPU resources—the more the CPU cores, the better the performance. Therefore, you should have more CPUs in the XCP server. We lab tested 128GB RAM and 48x core CPUs, which provided better performance than 8x CPUs and 8GB RAM.
- XCP copy with the `-parallel` option is based on the number of CPUs. The default number of parallel threads (seven) is sometimes sufficient for most XCP data transfer and migration operations. For XCP Windows by default, the number of parallel processes is equal to the number of CPUs. The maximum number for the `-parallel` option should be less than or equal to the number of cores.
- 10GbE is a good start for data transfer. However, we tested with 25GbE and 100GbE, which provided better data transfer and are recommended for large file-size data transfer.
- For Azure NetApp Files, the performance varies based on the service level. For more information, see the following table, which shows Azure NetApp Files service levels and performance details.
You must choose the right service level based on the throughput and workload types. Most customers start with the Premium level and change the service level based on the workload.

Customer scenarios

Overview

This section describes customer scenarios and their architectures.

Data lake to ONTAP NFS

This use case is based on the largest financial customer proof of concept (CPOC) that we have done. Historically, we used the NetApp In-Place Analytics Module (NIPAM) to move analytics data to NetApp ONTAP AI. However, because of recent enhancements and the improved performance of NetApp XCP as well as the unique NetApp data mover solution approach, we reran the data migration using NetApp XCP.

Customer challenges and requirements

Customer challenges and requirements that are worth noting include the following:

- Customers have different types of data, including structured, unstructured, and semistructured data, logs, and machine-to-machine data in data lakes. AI systems require all these types of data to process for prediction operations. When data is in a data lake-native file system, it is difficult to process.
- The customer’s AI architecture is not able to access data from Hadoop Distributed File System (HDFS) and Hadoop Compatible File System (HCFS), so the data is not available to AI operations. AI requires data in an understandable file system format such as NFS.
- Some special processes are required to move data from the data lake because of the large amount of data and high-throughput, and a cost-effective method is required to move the data to the AI system.

Data mover solution

In this solution, the MapR File System (MapR-FS) is created from local disks in the MapR cluster. The MapR NFS Gateway is configured on each data node with virtual IPs. The file server service stores and manages the MapR-FS data. NFS Gateway makes Map-FS data accessible from the NFS client through the virtual IP. An XCP instance is running on each MapR data node to transfer the data from the Map NFS Gateway to NetApp ONTAP NFS. Each XCP instance transfers a specific set of source folders to the destination location.

The following figure illustrates the NetApp data mover solution for MapR cluster using XCP.
For detailed customer use cases, recorded demos, and test results, see the Using XCP to Move Data from a Data Lake and High-Performance Computing to ONTAP NFS blog.

For detailed steps on moving MapR-FS data into ONTAP NFS by using NetApp XCP, see Appendix B in TR-4732: Big Data Analytics Data to Artificial Intelligence.

High-performance computing to ONTAP NFS

This use case is based on requests from field organizations. Some NetApp customers have their data in a high-performance computing environment, which provides data analytics for training models and enables research organizations to gain insight and understanding of large amount of digital data. NetApp field engineers need a detailed procedure to extract the data from IBM’s GPFS to NFS. We used NetApp XCP to migrate the data from GPFS to NFS so that GPUs can process the data. AI typically processes data from a network file system.

For more information about the high-performance computing to ONTAP NFS use case, a recorded demo, and test results, see the Using XCP to Move Data from a Data Lake and High-Performance Computing to ONTAP NFS blog.

For detailed steps on moving MapR-FS data into ONTAP NFS by using NetApp XCP, see Appendix A: GPFS to NFS—Detailed Steps in TR-4732: Big Data Analytics Data to Artificial Intelligence.

Using the XCP Data Mover to migrate millions of small files to flexible storage

This use case is based on the largest NetApp tourism industry customer for on-premises-to-cloud data migration. Because COVID-19 has reduced demand in the travel industry, customers want to save capital expenses on high-end storage in their on-premises environment for the demand pricing application. This customer has a tight SLA to migrate millions of small files to the cloud.

The following figure depicts data migration from on-premises to Azure NetApp Files for small files.
Using the XCP Data Mover to migrate large files

This use case is based on a television network customer. The customer wanted to migrate Oracle Recovery Manager (RMAN) backup files to the cloud and run the Oracle E-Business Suite (EBS) application by using Azure NetApp Files with Pacemaker software. The customer also wanted to migrate their database backup files to on-demand cloud storage and transfer large files (in the range of 25GB to 50GB each) to Azure.

The following figure illustrates the data migration from on-premises to Azure NetApp Files for large files.

For more information, see the NetApp XCP Data Mover Solution: On Premises to Cloud blog.

Duplicate files

NetApp received a request to find duplicate files from a single volume or multiple volumes. NetApp provided the following solution.

For single volume, run the following commands:
[root@mastr-51 linux]# ./xcp -md5 -match 'type==f and nlinks==1 and size != 0' 10.63.150.213:/common_volume/nfsconnector_hw_cert/ | sort | uniq -cd --check-chars=32

XCP 1.5; (c) 2020 NetApp, Inc.; Licensed to Calin Salagean [NetApp Inc] until Mon Dec 31 00:00:00 2029

176,380 scanned, 138,116 matched, 138,115 summed, 10 giants, 61.1 GiB in (763 MiB/s), 172 MiB out (2.57 MiB/s), 1m5s

Filtered: 38264 did not match
176,380 scanned, 138,116 matched, 138,116 summed, 10 giants, 62.1 GiB in (918 MiB/s), 174 MiB out (2.51 MiB/s), 1m9s.

3 00004964ca155eca1a71d0949c82e37e
nfsconnector_hw_cert/grid_01082017_174316/0/hadoopqe/accumulo/shell/pom.xml

2 000103fbed06d8071410c59047738389
nfsconnector_hw_cert/usr_hdp/2.5.3.0-37/hive2/doc/examples/files/dim-data.txt

2 000131053a46d67557d27bb678d5d4a1
nfsconnector_hw_cert/grid_01082017_174316/0/log/cluster/mahout_1/artifacts/classifier/20news_reduceddata/20news-bydate-test/alt.atheism/53265

For multiple volumes, run the following commands:

[root@mastr-51 linux]# cat multiplevolume_duplicate.sh
#!/usr/bin/bash

#user input
JUNCTION_PATHS='/nc_volume1 /nc_volume2 /nc_volume3 /oplogarchivevolume'
NFS_DATA_LIF='10.63.150.213'

#xcp operation
for i in $JUNCTION_PATHS
do
echo "start - $i" >> /tmp/duplicate_results
/usr/src/xcp/linux/xcp -md5 -match 'type==f and nlinks==1 and size != 0'
${NFS_DATA_LIF}:${i} | sort | uniq -cd --check-chars=32 | tee -a /tmp/duplicate_results
echo "end - $i" >> /tmp/duplicate_results
done

[root@mastr-51 linux]# nohup bash +x multiplevolume_duplicate.sh &
[root@mastr-51 linux]# cat /tmp/duplicate_results
Specific date-based scan and copy of data

This solution is based on a customer who needs to copy data based on a specific date. Verify the following details:

Created a file in Y: and checked the scan command to list them.

c:\XCP>dir Y:\karthik_test
Volume in drive Y is from
Volume Serial Number is 80F1-E201

Directory of Y:\karthik_test

05/26/2020 02:51 PM <DIR> .
05/26/2020 02:50 PM <DIR> ..
05/26/2020 02:51 PM             2,295 testfile.txt
   1 File(s)             2,295 bytes
   2 Dir(s)   658,747,392 bytes free

c:\XCP>

c:\XCP>xcp scan -match "strftime(ctime,'%Y-%m-%d')>'2020-05-01'" -fmt "'{},{}.format(iso(mtime),name)" Y:
XCP SMB 1.6; (c) 2020 NetApp, Inc.;Licensed to Calin Salagean [NetApp Inc] until Mon Dec 31 00:00:00 2029

It appears that you are not running XCP as Administrator. To avoid access issues please run XCP as Administrator.

2020-05-26_14:51:13.132465,testfile.txt
2020-05-26_14:51:00.074216,karthik_test

xcp scan -match strftime(ctime,'%Y-%m-%d')>'2020-05-01' -fmt '{},{}.format(iso(mtime),name) Y:\ : PASSED
30,205 scanned, 2 matched, 0 errors
Total Time : 4s
STATUS : PASSED

Copy the files based on date (2020 YearMay month first date) from Y: to Z:
c:\XCP>xcp copy -match "strftime(ctime,'%Y-%m-%d')>'2020-05-01'" Y: Z:dest_karthik
XCP SMB 1.6; (c) 2020 NetApp, Inc.; Licensed to Calin Salagean [NetApp Inc] until Mon Dec 31 00:00:00 2029

It appears that you are not running XCP as Administrator. To avoid access
issues please run XCP as Administrator.

30,205 scanned, 3 matched, 0 copied, 0 errors, 5s
xcp copy -match strftime(ctime,'%Y-%m-%d')>'2020-05-01' Y: Z:\dest_karthik : PASSED
30,205 scanned, 3 matched, 2 copied, 0 errors
Total Time : 6s
STATUS : PASSED

c:\XCP>

Check the destination Z:

c:\XCP>dir Z:\dest_karthik\karthik_test

Volume in drive Z is to
Volume Serial Number is 80F1-E202

Directory of Z:\dest_karthik\karthik_test

05/26/2020  02:51 PM    <DIR>          .
05/26/2020  02:50 PM    <DIR>          ..
05/26/2020  02:51 PM             2,295 testfile.txt
1 File(s)          2,295 bytes
2 Dir(s)     659,316,736 bytes free

c:\XCP>

Creating a CSV file from SMB/CIFS share

The following command dumps data in the CSV format. You can sum up the size column to get the total size of the data.

xcp scan -match "((now-x.atime) / 3600) > 31*day" -fmt "{}, {}, {}, {}
".format(relpath, name, strftime(x.atime, '%y-%m-%d-%H:%M:%S'),
humanize_size(size))" -preserve-atime >file.csv

The output should look similar to this example:

erase\report_av_fp_cdot_crosstab.csvreport_av_fp_cdot_crosstab.csv20-01-29-10:26:2449.6MiB

To scan up to the depth of three subdirectories and provide the result in sorting order, run the xcp -du command and dump the size at each directory level up to the depth of three subdirectories.
To sort, dump the information to a CSV file and sort the information.

This is a custom report that uses the `-fmt` command. It scans all the directories and dumps the name of the directory, path, and size of directory into a CSV file. You can sort the size column from the spreadsheet application.

**Data migration from 7-Mode to ONTAP**

This section provides detailed steps for migrating data from NetApp Data ONTAP operating in 7-Mode to ONTAP.

**Transitioning 7-Mode NFSv3 storage to ONTAP for NFS data**

This section provides the step-by-step procedure in the following table for transitioning a source 7-Mode NFSv3 export to an ONTAP system.

NetApp assumes that the source 7-Mode NFSv3 volume is exported and mounted on the client system and that XCP is already installed on a Linux system.

1. Verify that the target ONTAP system is healthy.
2. Verify that at least one nonroot aggregate exists on the target system. The aggregate is normal.
If there is no data aggregate, create a new one using the `storage aggr create` command.

3. Create a storage virtual machine (SVM) on the target cluster system.
CLUSTER::> vserver create -vserver dest -rootvolume dest_root -aggregate poc -rootvolume-security-style mixed
[Job 647] Job succeeded:
Vserver creation completed

Verify the security style and language settings of the source

Verify that the SVM was successfully created.

CLUSTER::> vserver show -vserver dest

Vserver: dest
    Vserver Type: data
    Vserver Subtype: default
    Vserver UUID: 91f6d786-0063-11e5-b114-00a09853a969
    Root Volume: dest_root
    Aggregate: poc
    NIS Domain: -
    Root Volume Security Style: mixed
    LDAP Client: -
    Default Volume Language Code: C.UTF-8
    Snapshot Policy: default
    Comment:
    Quota Policy: default
    List of Aggregates Assigned: -
    Limit on Maximum Number of Volumes allowed: unlimited
    Vserver Admin State: running
    Vserver Operational State: running
    Vserver Operational State Stopped Reason: -
    Allowed Protocols: nfs, cifs, fcp, iscsi, ndmp
    Disallowed Protocols: -
    Is Vserver with Infinite Volume: false
    QoS Policy Group: -
    Config Lock: false
    IPspace Name: Default

4. Remove the FCP, iSCSI, NDMP, and CIDS protocols from the target SVM.

CLUSTER::> vserver remove-protocols -vserver dest -protocols fcp,iscsi,ndmp,cifs

Verify that NFS is the allowed protocol for this SVM.
5. Create a new read-write data volume on the destination SVM. Verify that the security style, language settings, and capacity requirements match the source volume.

```bash
CLUSTER::> vol create -vserver dest -volume dest_nfs -aggregate poc
-size 150g -type RW -state online -security-style mixed
[Job 648] Job succeeded: Successful
```

6. Create a data LIF to serve NFS client requests.

```bash
CLUSTER::> network interface create -vserver dest -lif dest_lif -address
10.61.73.115 -netmask 255.255.255.0 -role data -data-protocol nfs -home
-node CLUSTER-01 -home-port e0l
```

Verify that the LIF was successfully created.

```bash
CLUSTER::> network interface show -vserver dest
Logical    Status     Network            Current
Current Is
Vserver     Interface  Admin/Oper Address/Mask       Node          Port
Home        ------- ------------------ --------------  ----------- ----
------------
------- ----
dest     dest_lif
true       up/up     10.61.73.113/24   CLUSTER-01   e0i
```

7. Create a static route with the SVM, if required.

```bash
CLUSTER::> network route create -vserver dest -destination 0.0.0.0/0
-gateway 192.168.100.111
```

Verify that the route was successfully created.
8. Mount the target NFS data volume in the SVM namespace.

```
CLUSTER::> volume mount -vserver dest -volume dest_nfs -junction-path /dest_nfs -active true
```

Verify that the volume is successfully mounted.

```
CLUSTER::> volume show -vserver dest -fields junction-path
vserver volume   junction-path
-------- -------- -------------
dest    dest_nfs /dest_nfs
dest    dest_root
```

2 entries were displayed.

You can also specify volume mount options (junction path) with the `volume create` command.

9. Start the NFS service on the target SVM.

```
CLUSTER::> vserver nfs start -vserver dest
```

Verify that the service is started and running.

```
CLUSTER::> vserver nfs status
The NFS server is running on Vserver "dest".
CLUSTER::> nfs show
Vserver: dest
    General Access: true
        v3: enabled
        v4.0: disabled
        4.1: disabled
        UDP: enabled
        TCP: enabled
    Default Windows User: -
    Default Windows Group: -
```
10. Verify that the default NFS export policy was applied to the target SVM.

```
CLUSTER::> vserver export-policy show -vserver dest
Vserver          Policy Name
------------------ -------------------
dest             default
```

11. If required, create a new custom export policy for the target SVM.

```
CLUSTER::> vserver export-policy create -vserver dest -policyname xcpexportpolicy
```

Verify that the new custom export policy was successfully created.

```
CLUSTER::> vserver export-policy show -vserver dest
Vserver          Policy Name
------------------ -------------------
dest             default
                xcpexportpolicy
2 entries were displayed.
```

12. Modify the export policy rules to allow access to NFS clients.

```
CLUSTER::> export-policy rule modify -vserver dest -ruleindex 1
         -policyname xcpexportpolicy -clientmatch 0.0.0.0/0 -rorule any -rwrule
         any -anon 0
Verify the policy rules have modified
CLUSTER::> export-policy rule show -instance
         Vserver: dest
         Policy Name: xcpexportpolicy
         Rule Index: 1
         Access Protocol: nfs3
         Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0
         RO Access Rule: none
         RW Access Rule: none
         User ID To Which Anonymous Users Are Mapped: 65534
         Superuser Security Types: none
         Honor SetUID Bits in SETATTR: true
         Allow Creation of Devices: true
```

13. Verify that the client is allowed access to the volume.
### CLUSTER::> export-policy check-access -vserver dest -volume dest_nfs
-client-ip 10.61.82.215 -authentication-method none -protocol nfs3
-access-type read-write

<table>
<thead>
<tr>
<th>Path</th>
<th>Policy</th>
<th>Policy Owner</th>
<th>Owner Type</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access</td>
<td>xcpexportpolicy</td>
<td>dest_root volume</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>read</td>
<td>xcpexportpolicy</td>
<td>dest_nfs volume</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>/dest_nfs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>read-write</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2 entries were displayed.

14. Connect to the Linux NFS server. Create a mount point for the NFS exported volume.

```bash
[root@localhost /]# cd /mnt
[root@localhost mnt]# mkdir dest
```

15. Mount the target NFSv3 exported volume at this mount point.

```
[root@localhost mnt]# mount -t nfs 10.61.73.115:/dest_nfs /mnt/dest
```

Verify that the mount point was successfully created.

```
[root@ localhost /]# mount | grep nfs
10.61.73.115:/dest_nfs on /mnt/dest type nfs
```

16. Create a test file on the NFS exported mount point to enable read-write access.
After the read-write test is complete, delete the file from the target NFS mount point.

17. Connect to the Linux client system in which XCP is installed. Browse to the XCP install path.

18. Query the source 7-Mode NFSv3 exports by running the `xcp show` command on the XCP Linux client host system.

```
[root@localhost dest]# touch test.txt
Verify the file is created
[root@localhost dest]# ls -l
total 0
-rw-r--r-- 1 root bin 0 Jun 2 03:16 test.txt
```

```
After the read-write test is complete, delete the file from the target NFS mount point.

17. Connect to the Linux client system in which XCP is installed. Browse to the XCP install path.

```
[root@localhost ~]# cd /linux/
[root@localhost linux]#
```

```
18. Query the source 7-Mode NFSv3 exports by running the `xcp show` command on the XCP Linux client host system.

```
[root@localhost]#/xcp show 10.61.82.215
== NFS Exports ==
Mounts  Errors  Server
  4       0  10.61.82.215
   Space   Files   Space   Files
   Free   Free     Used   Used Export
23.7 GiB  778,134  356 KiB       96 10.61.82.215:/vol/nfsvol1
17.5 GiB  622,463  1.46 GiB      117 10.61.82.215:/vol/nfsvol
328 GiB  10.8M   2.86 GiB       7,904 10.61.82.215:/vol/vol0/home
328 GiB  10.8M   2.86 GiB       7,904 10.61.82.215:/vol/vol0
```

```
== Attributes of NFS Exports ==
drwxr-xr-x --- root wheel 4KiB 4KiB 2d21h 10.61.82.215:/vol/nfsvol1
```

```
19. Scan the source NFSv3 exported paths and print the statistics of their file structure.

NetApp recommends putting the source NFSv3 exports in read-only mode during xcp scan, copy, and sync operations.
```
20. Copy the source 7-Mode NFSv3 exports to NFSv3 exports on the target ONTAP system.

```
[root@localhost /]# ./xcp copy 10.61.82.215:/vol/nfsvol
10.61.73.115:/dest_nfs
  44 scanned, 39 copied, 264 MiB in (51.9 MiB/s), 262 MiB out (51.5 MiB/s), 5s
  44 scanned, 39 copied, 481 MiB in (43.3 MiB/s), 479 MiB out (43.4 MiB/s), 10s
  44 scanned, 40 copied, 748 MiB in (51.2 MiB/s), 747 MiB out (51.3 MiB/s), 16s
  44 scanned, 40 copied, 1.00 GiB in (55.9 MiB/s), 1.00 GiB out (55.9 MiB/s), 21s
  44 scanned, 40 copied, 1.21 GiB in (42.8 MiB/s), 1.21 GiB out (42.8 MiB/s), 26s
Sending statistics...
  44 scanned, 43 copied, 1.46 GiB in (47.6 MiB/s), 1.45 GiB out (47.6 MiB/s), 31s.
```

21. After the copy is finished, verify that the source and destination NFSv3 exports have identical data. Run the xcp verify command.
If `xcp verify` finds differences between the source and destination data, then the error `no such file or directory` is reported in the summary. To fix that issue, run the `xcp sync` command to copy the source changes to the destination.

22. Before and during the cutover, run `verify` again. If the source has new or updated data, then perform incremental updates. Run the `xcp sync` command.

For this operation, the previous copy index name or number is required.

```
[root@localhost /]# ./xcp sync -id 3
Index: {source: '10.61.82.215:/vol/nfsvol', target: '10.61.73.115:/dest_nfs1'}
64 reviewed, 64 checked at source, 6 changes, 6 modifications, 51.7 KiB in (62.5 KiB/s), 22.7 KiB out (27.5 KiB/s), 0s.
xcp: sync '3': Starting search pass for 1 modified directory...
xcp: sync '3': Found 6 indexed files in the 1 changed directory
xcp: sync '3': Rereading the 1 modified directory to find what's new...
xcp: sync '3': Deep scanning the 1 directory that changed...
11 scanned, 11 copied, 12.6KiB in (6.19KiBps), 9.50 KiB out (4.66KiBps), 2s.
```

23. To resume a previously interrupted copy operation, run the `xcp resume` command.
After `resume` finishes copying files, run `verify` again so that the source and destination storage have identical data.

24. The NFSv3 client host needs to unmount the source NFSv3 exports provisioned from the 7-Mode storage and mounts the target NFSv3 exports from ONTAP. Cutover requires an outage.

**Transitioning 7-Mode volume Snapshot copies to ONTAP**

This section covers the procedure for transitioning a source 7-Mode volume NetApp Snapshot copy to ONTAP.
NetApp assumes that the source 7-Mode volume is exported and mounted on the client system and that XCP is already installed on a Linux system. A Snapshot copy is a point-in-time image of a volume that records incremental changes since the last Snapshot copy. Use the `-snap` option with a 7-Mode system as the source.

**Warning:** Keep the base Snapshot copy. Do not delete the base Snapshot copy after the baseline copy is complete. The base Snapshot copy is required for further sync operations.

1. Verify that the target ONTAP system is healthy.

```
CLUSTER::> cluster show
Node                  Health  Eligibility
--------------------- ------- ------------
CLUSTER-01            true    true
CLUSTER-02            true    true
2 entries were displayed.

CLUSTER::> node show
Node      Health Eligibility Uptime        Model       Owner    Location
--------- ------ ----------- ------------- ----------- --------
CLUSTER-01  true   true        78 days 21:01 FAS8060              RTP
CLUSTER-02  true   true        78 days 20:50 FAS8060              RTP
2 entries were displayed.

CLUSTER::> storage failover show
Takeover
Node           Partner        Possible State Description
-------------- -------------- --------
CLUSTER-01     CLUSTER-02     true     Connected to CLUSTER-02
CLUSTER-02     CLUSTER-01     true     Connected to CLUSTER-01
2 entries were displayed.
```

2. Verify that at least one nonroot aggregate exists on the target system. The aggregate is normal.
If there is no data aggregate, create a new one using the `storage aggr create` command.

3. Create an SVM on the target cluster system.
4. Remove the FCP, iSCSI, NDMP, and CIFS protocols from the target SVM.

```
CLUSTER::> vserver remove-protocols -vserver dest -protocols fcp,iscsi,ndmp,cifs
Verify that NFS is the allowed protocol for this SVM.
CLUSTER::> vserver show -vserver dest -fields allowed-protocols
```

```
-------- -----------------
dest    nfs
```

5. Create a new read-write data volume on the destination SVM. Verify that the security style, language settings, and capacity requirements match the source volume.

```
CLUSTER::> vol create -vserver dest -volume dest_nfs -aggregate poc -size 150g -type RW -state online -security-style mixed
[Job 648] Job succeeded: Successful
```

6. Create a data LIF to serve NFS client requests.

```
CLUSTER::> network interface create -vserver dest -lif dest_lif -address 10.61.73.115 -netmask 255.255.255.0 -role data -data-protocol nfs -home -node CLUSTER-01 -home-port e0i
```

Verify that the LIF was successfully created.

```
CLUSTER::> network interface show -vserver dest

Current Is
Vserver Interface Admin/Oper Address/Mask Node Port
Home ---------- ------- ------------------ -------------
--- ----
dest dest_lif up/up 10.61.73.113/24 CLUSTER-01 e0i
true
```

7. If required, create a static route with the SVM.

```
CLUSTER::> network route create -vserver dest -destination 0.0.0.0/0 -gateway 192.168.100.111
```

Verify that the route was successfully created.

```
CLUSTER::> network route show -vserver source

Vserver Destination Gateway Metric
------------------ --------------- -----
dest 0.0.0.0/0 10.61.73.1 20
```

8. Mount the target NFS data volume in the SVM namespace.
Verify that the volume was successfully mounted.

You can also specify the volume mount options (junction path) with the `volume create` command.

9. Start the NFS service on the target SVM.

Verify that the service is started and running.

10. Verify that the default NFS export policy is applied to the target SVM.
11. If required, create a new custom export policy for the target SVM.

   CLUSTER::> vserver export-policy create -vserver dest -policyname xcpexportpolicy

   Verify that the new custom export policy was successfully created.

   CLUSTER::> vserver export-policy show -vserver dest
   Vserver          Policy Name
   ---------------  -------------------
   dest             default
   dest             xcpexportpolicy
   2 entries were displayed.

12. Modify the export policy rules to allow access to NFS clients on the target system.

   CLUSTER::> export-policy rule modify -vserver dest -ruleindex 1
   -policyname xcpexportpolicy -clientmatch 0.0.0.0/0 -rorule any -rwrule any -anon 0
   Verify the policy rules have modified
   CLUSTER::> export-policy rule show -instance
   Vserver: dest
   Policy Name: xcpexportpolicy
   Rule Index: 1
   Access Protocol: nfs3
   Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0
     RO Access Rule: none
     RW Access Rule: none
   User ID To Which Anonymous Users Are Mapped: 65534
     Superuser Security Types: none
     Honor SetUID Bits in SETATTR: true
     Allow Creation of Devices: true

13. Verify that the client has access to the target volume.
14. Connect to the Linux NFS server. Create a mount point for the NFS exported volume.

```bash
[root@localhost /]# cd /mnt
[root@localhost mnt]# mkdir dest
```

15. Mount the target NFSv3 exported volume at this mount point.

```
[root@localhost mnt]# mount -t nfs 10.61.73.115:/dest_nfs /mnt/dest
```

Verify that the mount point was successfully created.

```
[root@ localhost /]# mount | grep nfs
10.61.73.115:/dest_nfs on /mnt/dest type nfs
```

16. Create a test file on the NFS exported mount point to enable read-write access.

```
[root@localhost dest]# touch test.txt
Verify the file is created
[root@localhost dest]# ls -l
total 0
-rw-r--r-- 1 root bin 0 Jun 2 03:16 test.txt
```
After the read-write test is complete, delete the file from the target NFS mount point.

17. Connect to the Linux client system in which XCP is installed. Browse to the XCP install path.

```
[root@localhost ~]# cd /linux/
[root@localhost linux]#
```

18. Query the source 7-Mode NFSv3 exports by running the `xcp show` command on the XCP Linux client host system.

```
[root@localhost ]#./xcp show 10.61.82.215
== NFS Exports ==
Mounts  Errors  Server
  4     0  10.61.82.215
Space   Files    Space   Files
  23.7 GiB 778,134  356 KiB   96 10.61.82.215:/vol/nfsvoll
  17.5 GiB 622,463  1.46 GiB 117 10.61.82.215:/vol/nfsvol
  328 GiB  10.8M  2.86 GiB  8704 10.61.82.215:/vol/vol0/home
  328 GiB  10.8M  2.86 GiB  8704 10.61.82.215:/vol/vol0
== Attributes of NFS Exports ==
drwxr-xr-x --- root wheel 4KiB 4KiB 2d21h 10.61.82.215:/vol/nfsvoll
```

19. Scan the source NFSv3 exported paths and print the statistics of their file structure.

NetApp recommends putting the source NFSv3 exports in read-only mode during `xcp` `scan`, `copy`, and `sync` operations. In `sync` operation, you must pass the `-snap` option with a corresponding value.
20. Copy the source 7-Mode NFSv3 snapshot (base) to NFSv3 exports on the target ONTAP system.

```
[root@localhost /]# /xcp copy 10.61.82.215:/vol/nfsvol/.snapshot/snap1 10.61.73.115:/dest_nfs
44 scanned, 39 copied, 264 MiB in (51.9 MiB/s), 262 MiB out (51.5 MiB/s), 5s
44 scanned, 39 copied, 481 MiB in (43.3 MiB/s), 479 MiB out (43.4 MiB/s), 10s
44 scanned, 40 copied, 748 MiB in (51.2 MiB/s), 747 MiB out (51.3 MiB/s), 16s
44 scanned, 40 copied, 1.00 GiB in (55.9 MiB/s), 1.00 GiB out (55.9 MiB/s), 21s
44 scanned, 40 copied, 1.21 GiB in (42.8 MiB/s), 1.21 GiB out (42.8 MiB/s), 26s
Sending statistics...
44 scanned, 43 copied, 1.46 GiB in (47.6 MiB/s), 1.45 GiB out (47.6 MiB/s), 31s.
```

Keep this base snapshot for further sync operations.

21. After copy is complete, verify that the source and destination NFSv3 exports have identical data. Run the `xcp verify` command.
If `verify` finds differences between the source and destination data, then the error `no such file or directory` is reported in the summary. To fix that issue, run the `xcp sync` command to copy the source changes to the destination.

22. Before and during the cutover, run `verify` again. If the source has new or updated data, then perform incremental updates. If there are incremental changes, create a new Snapshot copy for these changes and pass that snapshot path with the `-snap` option for sync operations.

Run the `xcp sync` command with the `-snap` option and snapshot path.

```
[root@localhost /]# ./xcp verify 10.61.82.215:/vol/nfsvol
10.61.73.115:/dest_nfs
44 scanned, 44 found, 28 compared, 27 same data, 2.41 GiB in (98.4 MiB/s), 6.25 MiB out (255 KiB/s), 26s
44 scanned, 44 found, 30 compared, 29 same data, 2.88 GiB in (96.4 MiB/s), 7.46 MiB out (249 KiB/s), 31s
44 scanned, 100% found (43 have data), 43 compared, 100% verified (data, attrs, mods), 2.90 GiB in (92.6 MiB/s), 7.53 MiB out (240 KiB/s), 32s.
```

For this operation, the base snapshot is required.

23. To resume a previously interrupted copy operation, run the `xcp resume` command.

```
[root@localhost /]# ./xcp sync -id 3
Index: {source: '10.61.82.215:/vol/nfsvol/.snapshot/snap1', target: '10.61.73.115:/dest_nfs1'}
64 reviewed, 64 checked at source, 6 changes, 6 modifications, 51.7 KiB in (62.5 KiB/s), 22.7 KiB out (27.5 KiB/s), 0s.
xcp: sync '3': Starting search pass for 1 modified directory...
xcp: sync '3': Found 6 indexed files in the 1 changed directory
xcp: sync '3': Rereading the 1 modified directory to find what's new...
xcp: sync '3': Deep scanning the 1 directory that changed...
11 scanned, 11 copied, 12.6 KiB in (6.19 KiB/s), 9.50 KiB out (4.66 KiB/s), 2s..
```

653
24. The NFSv3 client host must unmount the source NFSv3 exports provisioned from the 7-Mode storage and mount the target NFSv3 exports from ONTAP. This cutover requires an outage.

Migrating ACLv4 from NetApp 7-Mode to a NetApp storage system

This section covers the step-by-step procedure for transitioning a source NFSv4 export to an ONTAP system.

NetApp assumes that the source NFSv4 volume is exported and mounted on the client system and that XCP is already installed on a Linux system. The source should be a NetApp 7-Mode system that supports ACLs. ACL migration is supported from NetApp to NetApp only. To copy files with a special character in the name, make sure the source and destination support UTF-8 encoded language.

Prerequisites for migrating a source NFSv4 export to ONTAP

Before you migrate a source NFSv4 export to ONTAP, the following prerequisites must be met:

- The destination system must have NFSv4 configured.
- The NFSv4 source and target must be mounted on the XCP host. Select NFS v4.0 to match the source and target storage and verify that the ACLs are enabled on the source and target system.
- XCP requires the source/target path to be mounted on the XCP host for ACL processing. In the following example, vol1(10.63.5.56:/vol1) is mounted on the /mnt/vol1 path:
```
[root@localhost ~]# df -h
Filesystem Size  Used
Avail Use% Mounted on
10.63.5.56:/vol1 973M 4.2M
969M 1% /mnt/vol1
[root@localhost ~]# ./xcp scan -l -acl4 10.63.5.56:/vol1/
XCP <version>; (c) 2020 NetApp, Inc.; Licensed to XXX [NetApp Inc] until Sun Mar 31 00:00:00 2029
```

Subdirectories options

The two options to work with subdirectories are as follows:

- For XCP to work on a subdirectory (/vol1/DIR1/DIR11), mount the complete path (10.63.5.56:/vol1/DIR1/DIR11) on the XCP host.

  If the complete path is not mounted, XCP reports the following error:

```
[root@localhost ~]# ./xcp scan -l -acl4 10.63.5.56:/vol1/DIR1/DIR11
XCP <version>; (c) 2020 NetApp, Inc.; Licensed to XXX [NetApp Inc] until Sun Mar 31 00:00:00 2029
xcp: ERROR: For xcp to process ACLs, please mount 10.63.5.56:/vol1/DIR1/DIR11 using the OS nfs4 client.
```

- Use the subdirectory syntax (mount: subdirectory/qtrees/snapshot), as shown in the example below:
Complete the following steps to migrate ACLv4 from NetApp 7-Mode to a NetApp storage system.

1. Verify that the target ONTAP system is healthy.

```
CLUSTER::> cluster show
Node                  Health  Eligibility
--------------------- ------- ------------
CLUSTER-01            true    true
CLUSTER-02            true    true
2 entries were displayed.

CLUSTER::> node show
Node      Health Eligibility Uptime        Model       Owner    Location
--------- ------ ----------- ------------- ----------- --------
--------- ------ ----------- ----------- --------- --------
CLUSTER-01 true   true        78 days 21:01 FAS8060  RTP
CLUSTER-02 true   true        78 days 20:50 FAS8060  RTP
2 entries were displayed.

CLUSTER::> storage failover show
Takeover
Node           Partner        Possible State Description
-------------- -------------- -------------------------------
CLUSTER-01     CLUSTER-02     true     Connected to CLUSTER-02
CLUSTER-02     CLUSTER-01     true     Connected to CLUSTER-01
2 entries were displayed.
```

2. Verify that at least one nonroot aggregate exists on the target system. The aggregate is normal.
If there is no data aggregate, create a new one using the `storage aggr create` command.

3. Create an SVM on the target cluster system.

```
CLUSTER::> vserver create -vserver dest -rootvolume dest_root -aggregate poc -rootvolume-security-style mixed
[Job 647] Job succeeded:
Vserver creation completed
```

Verify that the SVM was successfully created.
4. Remove the FCP, iSCSI, NDMP, and CIFS protocols from the target SVM.

```
CLUSTER::> vserver remove-protocols -vserver dest -protocols fcp,iscsi,ndmp,cifs
```

Verify that NFS is the allowed protocol for this SVM.

```
CLUSTER::> vserver show -vserver dest -fields allowed-protocols
vserver allowed-protocols
------- -----------------
dest    nfs
```

5. Create a new read-write data volume on the destination SVM. Verify that the security style, language settings, and capacity requirements match the source volume.
6. Create a data LIF to serve NFS client requests.

```
CLUSTER::> network interface create -vserver dest -lif dest_lif -address 10.61.73.115 -netmask 255.255.255.0 -role data -data-protocol nfs -home -node CLUSTER-01 -home-port e01
```

Verify that the LIF was successfully created.

```
CLUSTER::> network interface show -vserver dest

Logical Status Network Current Is
Vserver Interface Admin/Oper Address/Mask       Node          Port
Home
--- ------ ------------------ -------------- ------- ----
dest    dest_lif        up/up  10.61.73.113/24   CLUSTER-01  e01
true
```

7. If required, create a static route with the SVM.

```
CLUSTER::> network route create -vserver dest -destination 0.0.0.0/0 -gateway 192.168.100.111
```

Verify that the route was successfully created.

```
CLUSTER::> network route show -vserver source

Vserver Destination Gateway Metric
------------------------ ------------------------ ----
dest  0.0.0.0/0  10.61.73.1  20
```

8. Mount the target NFS data volume in the SVM namespace.
CLUSTER::> volume mount -vserver dest -volume dest_nfs -junction-path /dest_nfs -active true

Verify that the volume was successfully mounted.

```
CLUSTER::> volume show -vserver dest -fields junction-path
vserver volume   junction-path
------- -------- -------------
dest    dest_nfs /dest_nfs
dest    dest_root
/
2 entries were displayed.
```

You can also specify the volume mount options (junction path) with the `volume create` command.

9. Start the NFS service on the target SVM.

```
CLUSTER::> vserver nfs start -vserver dest
```

Verify that the service is started and running.

```
CLUSTER::> vserver nfs status
The NFS server is running on Vserver "dest".
CLUSTER::> nfs show
Vserver: dest
    General Access: true
    v3: enabled
    v4.0: enabled
    4.1: disabled
    UDP: enabled
    TCP: enabled
    Default Windows User: -
    Default Windows Group: -
```

10. Check that the default NFS export policy is applied to the target SVM.

```
CLUSTER::> vserver export-policy show -vserver dest
Vserver          Policy Name
---------------  -------------------
dest             default
```

660
11. If required, create a new custom export policy for the target SVM.

```
CLUSTER::> vserver export-policy create -vserver dest -policyname xcpexportpolicy
```

Verify that the new custom export policy was successfully created.

```
CLUSTER::> vserver export-policy show -vserver dest
Vserver          Policy Name
---------------  -------------------
dest             default
dest             xcpexportpolicy
2 entries were displayed.
```

12. Modify the export policy rules to allow access to NFS clients.

```
CLUSTER::> export-policy rule modify -vserver dest -ruleindex 1
-policyname xcpexportpolicy -clientmatch 0.0.0.0/0 -rorule any -rwrule any
-anon 0
```

Verify that the policy rules have been modified.

```
CLUSTER::> export-policy rule show -instance
Vserver: dest
Policy Name: xcpexportpolicy
Rule Index: 1
Access Protocol: nfs3
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0
RO Access Rule: none
RW Access Rule: none
User ID To Which Anonymous Users Are Mapped: 65534
Superuser Security Types: none
Honor SetUID Bits in SETATTR: true
Allow Creation of Devices: true
```

13. Verify that the client is allowed access to the volume.
14. Connect to the Linux NFS server. Create a mount point for the NFS exported volume.

[root@localhost /]# cd /mnt
[root@localhost mnt]# mkdir dest

15. Mount the target NFSv4 exported volume at this mount point.

The NFSv4 volumes should be exported but not necessarily mounted by the NFS server. If they can be mounted, the XCP Linux host client mounts these volumes.

[root@localhost mnt]# mount -t nfs4 10.63.5.56:/vol1 /mnt/vol1

Verify that the mount point was successfully created.

[root@localhost mnt]# mount | grep nfs
10.63.5.56:/vol1 on /mnt/vol1 type nfs4
(rw,relatime,vers=4.0,rsize=65536,wsize=65536,namlen=255,hard,proto=tcp, timeo=600,
retrans=2,sec=sys,clientaddr=10.234.152.84,local_lock=none,addr=10.63.5.56)

16. Create a test file on the NFS exported mount point to enable read-write access.

[root@localhost dest]# touch test.txt
Verify the file is created.

```
[root@localhost dest]# ls -l
total 0
-rw-r--r-- 1 root bin 0 Jun 2 03:16 test.txt
```

ℹ️ After the read-write test is complete, delete the file from the target NFS mount point.

17. Connect to the Linux client system in which XCP is installed. Browse to the XCP install path.

```
[root@localhost ~]# cd /linux/
[root@localhost linux]#
```

18. Query the source NFSv4 exports by running the `xcp show` command on the XCP Linux client host system.
19. Scan the source NFSv4 exported paths and print the statistics of their file structure.

NetApp recommends putting the source NFSv4 exports in read-only mode during xcp scan, copy, and sync operations.

[root@localhost]# ./xcp scan -acl4 10.63.5.56:/vol1
XCP <version>; (c) 2020 NetApp, Inc.; Licensed to xxx [NetApp Inc] until
Mon Dec 31 00:00:00 2029
vol1
vol1/test/f1
vol1/test
3 scanned, 3 getaclsls, 3 v3perms, 1.59 KiB in (1.72 KiB/s), 696 out
(753/s), 0s.
20. Copy source NFSv4 exports to NFSv4 exports on the target ONTAP system.

```
[root@localhost]# ./xcp copy -acl4 -newid id1 10.63.5.56:/vol1
10.63.5.56:/vol2
XCP <version>; (c) 2020 NetApp, Inc.; Licensed to xxx [NetApp Inc] until
Mon Dec 31 00:00:00 2029
3 scanned, 2 copied, 3 indexed, 3 getacls, 3 v3perms, 1 setacl, 14.7 KiB
in (11.7 KiB/s), 61 KiB out (48.4 KiB/s), 1s..
```

21. After copy is complete, verify that the source and destination NFSv4 exports have identical data. Run the xcp verify command.

```
[root@localhost]# ./xcp verify -acl4 -noid 10.63.5.56:/vol1
10.63.5.56:/vol2
XCP <version>; (c) 2020 NetApp, Inc.; Licensed to xxx [NetApp Inc] until
Mon Dec 31 00:00:00 2029
3 scanned, 100% found (0 have data), 100% verified (data, attrs, mods,
acls), 6 getacls, 6 v3perms, 2.90 KiB in (4.16 KiB/s), 2.94 KiB out
(4.22 KiB/s), 0s.
```

If verify finds differences between the source and destination data, then the error no such file or directory is reported in the summary. To fix that issue, run the xcp sync command to copy the source changes to the destination.

22. Before and during the cutover, run verify again. If the source has new or updated data, then perform incremental updates. Run the xcp sync command.

```
[root@root@localhost]# ./xcp sync -id id1
XCP <version>; (c) 2020 NetApp, Inc.; Licensed to xxx [NetApp Inc] until
Mon Dec 31 00:00:00 2029
xcp: Index: {source: 10.63.5.56:/vol1, target: 10.63.5.56:/vol2}
3 reviewed, 3 checked at source, no changes, 3 reindexed, 25.6 KiB in
(32.3 KiB/s), 23.3 KiB out (29.5 KiB/s), 0s.
```

For this operation, the previous copy index name or number is required.

23. To resume a previously interrupted copy operation, run the xcp resume command.
After `resume` finishes copying files, run `verify` again so that the source and destination storage have identical data.

**Transitioning 7-Mode SMB storage to ONTAP for CIFS data**

This section covers the step-by-step procedure for transitioning a source 7-Mode SMB share to an ONTAP system.

NetApp assumes that the 7-Mode and ONTAP systems are SMB licensed. The destination SVM is created, the source and destination SMB shares are exported, and XCP is installed and licensed.

1. Scan the SMB shares for the files and directories.
2. Copy the files (with or without ACL) from the source to the destination SMB share. The following example shows a copy with ACL.
C:\xcp\xcp copy -acl -fallback-user "DOMAIN\gabi" -fallback-group "DOMAIN\Group" \10.61.77.189\performance_SMB_home_dirs \10.61.77.56\performance_SMB_home_dirs
XCP SMB 1.6; (c) 2020 NetApp, Inc.; Licensed to xxxx xxxx[NetApp Inc] until Mon Dec 31 00:00:00 2029
7,898 scanned, 0 errors, 0 skipped, 184 copied, 96.1MiB (19.2MiB/s), 5s
7,898 scanned, 0 errors, 0 skipped, 333 copied, 519MiB (84.7MiB/s), 10s
7,898 scanned, 0 errors, 0 skipped, 366 copied, 969MiB (89.9MiB/s), 15s
7,898 scanned, 0 errors, 0 skipped, 422 copied, 1.43GiB (99.8MiB/s), 20s
7,898 scanned, 0 errors, 0 skipped, 1,100 copied, 1.69GiB (52.9MiB/s), 25s
7,898 scanned, 0 errors, 0 skipped, 1,834 copied, 1.94GiB (50.4MiB/s), 30s
7,898 scanned, 0 errors, 0 skipped, 1,906 copied, 2.43GiB (100MiB/s), 35s
7,898 scanned, 0 errors, 0 skipped, 2,937 copied, 2.61GiB (36.6MiB/s), 40s
7,898 scanned, 0 errors, 0 skipped, 2,969 copied, 3.09GiB (100.0MiB/s), 45s
7,898 scanned, 0 errors, 0 skipped, 3,001 copied, 3.58GiB (100.0MiB/s), 50s
7,898 scanned, 0 errors, 0 skipped, 3,298 copied, 4.01GiB (88.0MiB/s), 55s
7,898 scanned, 0 errors, 0 skipped, 5,614 copied, 4.01GiB (679KiB/s), 1m0s
7,898 scanned, 0 errors, 0 skipped, 7,879 copied, 4.02GiB (445KiB/s), 1m5s
7,898 scanned, 0 errors, 0 skipped, 7,897 copied, 4.02GiB (63.2MiB/s), 1m5s

If there is no data aggregate, create a new one using the storage aggr create command.

3. Sync the files on the source and destination.

C:\xcp\xcp sync -acl -fallback-user "DOMAIN\gabi" -fallback-group "DOMAIN\Group" \10.61.77.189\performance_SMB_home_dirs \10.61.77.56\performance_SMB_home_dirs
XCP SMB 1.6; (c) 2020 NetApp, Inc.; Licensed to xxxx xxxx[NetApp Inc] until Mon Dec 31 00:00:00 2029
10,796 scanned, 4,002 compared, 0 errors, 0 skipped, 0 copied, 0 removed, 5s
15,796 scanned, 8,038 compared, 0 errors, 0 skipped, 0 copied, 0 removed, 10s
15,796 scanned, 8,505 compared, 0 errors, 0 skipped, 0 copied, 0
4. Verify that the files were copied correctly.
CIFS data migration with ACLs from a source storage box to ONTAP

This section covers the step-by-step procedure for migrating CIFS data with security information from a source to a target ONTAP system.

1. Verify that the target ONTAP system is healthy.
2. Verify that at least one nonroot aggregate exists on the target system. The aggregate is normal.
cluster::*> storage aggregate show

<table>
<thead>
<tr>
<th>Aggregate</th>
<th>Size</th>
<th>Available</th>
<th>Used%</th>
<th>State</th>
<th>#Vols</th>
<th>Nodes</th>
<th>RAID</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggr0_sti96_vsim_ucs540o</td>
<td>7.58GB</td>
<td>373.3MB</td>
<td>95%</td>
<td>online</td>
<td>1</td>
<td>sti96-vsim-raid_dp, ucs540o</td>
<td>normal</td>
<td></td>
</tr>
<tr>
<td>aggr0_sti96_vsim_ucs540p</td>
<td>7.58GB</td>
<td>373.3MB</td>
<td>95%</td>
<td>online</td>
<td>1</td>
<td>sti96-vsim-raid_dp, ucs540p</td>
<td>normal</td>
<td></td>
</tr>
<tr>
<td>aggr_001</td>
<td>103.7GB</td>
<td>93.63GB</td>
<td>10%</td>
<td>online</td>
<td>1</td>
<td>sti96-vsim-raid_dp, ucs540p</td>
<td>normal</td>
<td></td>
</tr>
<tr>
<td>sti96_vsim_ucs540o_aggr1</td>
<td>23.93GB</td>
<td>23.83GB</td>
<td>0%</td>
<td>online</td>
<td>1</td>
<td>sti96-vsim-raid_dp, ucs540o</td>
<td>normal</td>
<td></td>
</tr>
<tr>
<td>sti96_vsim_ucs540p_aggr1</td>
<td>23.93GB</td>
<td>23.93GB</td>
<td>0%</td>
<td>online</td>
<td>0</td>
<td>sti96-vsim-raid_dp, ucs540p</td>
<td>normal</td>
<td></td>
</tr>
</tbody>
</table>

5 entries were displayed.

ℹ️ If there is no data aggregate, create a new one using the `storage aggr create` command.

3. Create an SVM on the target cluster system.
cluster::*> vserver create -vserver vs1 -rootvolume root_vs1 -aggregate sti96_vsim_ucs540o_aggr1 -rootvolume-security-style mixed

Verify that the SVM was successfully created.
C2_sti96-vsim-ucs540o_cluster::*>  vserver show -vserver vs1
Vserver: vs1
  Vserver Type: data
  Vserver Subtype: default
  Vserver UUID: f8bc54be-d91b-11e9-b99c-005056a7e57e
  Root Volume: root_vs1
  Aggregate: sti96_vsim_ucs540o_aggr1
  NIS Domain: NSQA-RTP-NIS1
  Root Volume Security Style: mixed
  LDAP Client: esisconfig
  Default Volume Language Code: C.UTF-8
  Snapshot Policy: default
  Data Services: data-nfs, data-cifs,
  data-flexcache, data-iscsi
  Comment: vs1
  Quota Policy: default
  List of Aggregates Assigned: -
  Limit on Maximum Number of Volumes allowed: unlimited
  Vserver Admin State: running
  Vserver Operational State: running
  Vserver Operational State Stopped Reason: -
  Allowed Protocols: nfs, cifs, fcp, iscsi, ndmp
  Disallowed Protocols: -
  Is Vserver with Infinite Volume: false
  QoS Policy Group: -
  Caching Policy Name: -
  Config Lock: false
  Volume Delete Retention Period: 0
  IPspace Name: Default
  Foreground Process: -
  Is Msid Preserved for DR: false
  Force start required to start Destination in muliple IDP fan-out case: false
  Logical Space Reporting: false
  Logical Space Enforcement: false

4. Create a new read-write data volume on the destination SVM. Verify that the security style, language settings, and capacity requirements match the source volume.
5. Create a data LIF to serve SMB client requests.

```
cluster::> vol create -vserver vs1 -volume dest_vol -aggregate aggr_001 -size 150g type RW -state online -security-style ntfs
```

Verify that the LIF was successfully created.

```
cluster::*> network interface create -vserver vs1 -lif sti96-vsim-ucs540o_data1 -address 10.237.165.87 -netmask 255.255.240.0 -role data -data-protocol nfs,cifs -home-node sti96-vsim-ucs540o -home-port e0d
```

```
cluster::*> network interface show -vserver vs1

Current Is
Vserver     Interface  Admin/Oper Address/Mask       Node          Port
Home
----------- ---------- ------------------ -------------- ----
vs1
sti96-vsim-ucs540o_data1
    up/up    10.237.165.87/20   sti96-vsim-ucs540o  e0d
true
```

6. If required, create a static route with the SVM.

```
Network route create -vserver dest -destination 0.0.0.0/0 -gateway 10.237.160.1
```

Verify that the route was successfully created.

```
cluster::*> network route show -vserver vs1

Vserver Destination Gateway      Metric
------------------- --------------- ----- ----
vs1
    0.0.0.0/0       10.237.160.1    20
    ::/0            fd20:8b1e:b255:9155::1   20
2 entries were displayed.
```

7. Mount the target data volume in the SVM namespace.
Verify that the volume is successfully mounted.

Note: You can also specify the volume mount options (junction path) with the volume create command.

8. Start the CIFS service on the target SVM.

Verify that the service is started and running.

If required, create a new custom export policy for the target SVM.
10. Modify the export policy rules to allow access to CIFs clients.

```
CLUSTER::> vserver export-policy create -vserver vs1 -policyname xcpexport
```

```
CLUSTER::> export-policy rule modify -vserver dest -ruleindex 1 -policyname xcpexportpolicy -clientmatch 0.0.0.0/0 -rorule any -rwrule any -anon 0
```

Verify that the policy rules are modified.
cluster:*> export-policy rule show -instance
   Vserver: vs1
   Policy Name: default
   Rule Index: 1
   Access Protocol: any
List of Client Match Hostnames, IP Addresses, Netgroups, or Domains: 0.0.0.0/0
   RO Access Rule: any
   RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 65534
   Superuser Security Types: any
   Honor SetUID Bits in SETATTR: true
   Allow Creation of Devices: true
   NTFS Unix Security Options: fail
   Vserver NTFS Unix Security Options: use_export_policy
   Change Ownership Mode: restricted
   Vserver Change Ownership Mode: use_export_policy
   Policy ID: 12884901889
   Vserver: vs1
   Policy Name: default
   Rule Index: 2
   Access Protocol: any
List of Client Match Hostnames, IP Addresses, Netgroups, or Domains: 0:0:0:0:0:0:0:0/0
   RO Access Rule: any
   RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 65534
   Superuser Security Types: none
   Honor SetUID Bits in SETATTR: true
   Allow Creation of Devices: true
   NTFS Unix Security Options: fail
   Vserver NTFS Unix Security Options: use_export_policy
   Change Ownership Mode: restricted
   Vserver Change Ownership Mode: use_export_policy
   Policy ID: 12884901889
2 entries were displayed.

11. Verify that the client is allowed access to the volume.
cluster::*> export-policy check-access -vserver vs1 -volume dest_vol -client-ip 10.234.17.81 -authentication-method none -protocol cifs -access-type read-write

<table>
<thead>
<tr>
<th>Path</th>
<th>Policy</th>
<th>Owner</th>
<th>Owner Type</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>default</td>
<td>root_vs1</td>
<td>volume</td>
<td>1</td>
</tr>
<tr>
<td>read</td>
<td>default</td>
<td>dest_vol</td>
<td>volume</td>
<td>1</td>
</tr>
<tr>
<td>/dest_vol</td>
<td>default</td>
<td>dest_vol</td>
<td>volume</td>
<td>1</td>
</tr>
<tr>
<td>read-write</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2 entries were displayed.

12. Connect to the Windows client system where XCP is installed. Browse to the XCP install path.

C:\WRSKDNT>dir c:\netapp\xcp
C:\netapp\xcp
  Volume in drive C has no label.
  Volume Serial Number is 5C04-C0C7
Directory of c:\netapp\xcp
  09/18/2019  09:30 AM    <DIR>          .
  09/18/2019  09:30 AM    <DIR>          ..
  06/25/2019  06:27 AM               304 license
  09/18/2019  09:30 AM    <DIR>          Logs
  09/29/2019  08:45 PM        12,143,105 xcp.exe
      2 File(s)        12,143,409 bytes
      3 Dir(s)       29,219,549,184 bytes free

13. Query the source node SMB exports by running the `xcp show` command on the XCP Windows client host system.
14. Run the `help` command for copy.
C:\WRS HDNT>c:\netapp\xcp\xcp help copy
c:\netapp\xcp\xcp help copy
XCP SMB 1.6; (c) 2020 NetApp, Inc.; Licensed to XXX [NetApp Inc] until
Mon Dec 31 00:00:00 2029
usage: xcp copy [-h] [-v] [-parallel <n>] [-match <filter>] [-preserve-
atime]
[-acl] [-fallback-user FALLBACK_USER]
[-fallback-group FALLBACK_GROUP] [-root]
source target
positional arguments:
  source
  target
optional arguments:
  -h, --help            show this help message and exit
  -v                    increase debug verbosity
  -parallel <n>         number of concurrent processes (default: <cpu-
count>)
  -match <filter>       only process files and directories that match
                        the
                        filter (see `xcp help -match` for details)
  -preserve-atime       restore last accessed date on source
  -acl                  copy security information
  -fallback-user FALLBACK_USER
                        the name of the user on the target machine to
                        receive
                        the permissions of local (non-domain) source
                        machine
                        users (eg. domain\administrator)
  -fallback-group FALLBACK_GROUP
                        the name of the group on the target machine to
                        receive
                        the permissions of local (non-domain) source
                        machine
                        groups (eg. domain\administrators)
  -root                 copy acl for root directory

15. On the target ONTAP system, get the list of local user and local group names that you need to provide as
values for the fallback-user and fallback-group arguments path.
16. To migrate the CIFS data with ACLs from the source to target, run the `xcp copy` command with the `-acl` and `-fallback-user/group` options.

For the `fallback-user/group` options, specify any user or group that can be found in Active Directory or local user/group to target system.
If `xcp copy` results in the error message `ERROR failed to obtain fallback security principal`, add the destination box in the hosts file (`C:\Windows\System32\drivers\etc\hosts`).
Use the following format for the NetApp storage destination box entry.

<data vserver data interface ip> 1 or more white spaces <cifs server name>

cluster::*> cifs show

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Name</th>
<th>Admin</th>
<th>Name</th>
<th>Style</th>
</tr>
</thead>
<tbody>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-------</td>
<td>------------------</td>
<td>-------</td>
</tr>
<tr>
<td>vs1</td>
<td>D60AB15C2AFC4D6</td>
<td>up</td>
<td>CTL</td>
<td>domain</td>
</tr>
</tbody>
</table>

C2_sti96-vsim-ucs540o_cluster::*> network interface show

<table>
<thead>
<tr>
<th>Logical Cluster</th>
<th>Status</th>
<th>Network</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>sti96-vsim-ucs540p_clus1</td>
<td>up/up</td>
<td>192.168.148.136/24</td>
<td>sti96-vsim-ucs540p_e0a</td>
</tr>
<tr>
<td>true</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sti96-vsim-ucs540p_clus2</td>
<td>up/up</td>
<td>192.168.148.137/24</td>
<td>sti96-vsim-ucs540p_e0b</td>
</tr>
<tr>
<td>true</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vs1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sti96-vsim-ucs540o_data1</td>
<td>up/up</td>
<td>10.237.165.87/20</td>
<td>sti96-vsim-ucs540o_e0d</td>
</tr>
<tr>
<td>true</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sti96-vsim-ucs540o_data1_inet6</td>
<td>up/up</td>
<td>fd20:8b1e:b255:9155::583/64</td>
<td>sti96-vsim-ucs540o_e0d</td>
</tr>
<tr>
<td>true</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sti96-vsim-ucs540o_data2</td>
<td>up/up</td>
<td>10.237.165.88/20</td>
<td>sti96-vsim-ucs540o_e0e</td>
</tr>
<tr>
<td>true</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10.237.165.87  D60AB15C2AFC4D6  -> destination box entry to be added in hosts file.

18. If you still get the error message ERROR failed to obtain fallback security principal after adding the destination box entry in the hosts files, then the user/group does not exist in the target system.
19. Use `xcp copy` to migrate CIFs data with ACLs (with or without the root folder).

Without the root folder, run the following commands:
C:\WRSHDNT>c:\netapp\xcp\xcp copy -acl -fallback-user \
D60AB15C2AFC4D6\Administrator -fallback-group BUILTIN\Users \
10.237.165.79\source_share 10.237.165.89\dest_share 
C:\netapp\xcp\xcp copy -acl -fallback-user \
D60AB15C2AFC4D6\Administrator -fallback-group BUILTIN\Users \
10.237.165.79\source_share 10.237.165.89\dest_share 
XCP SMB 1.6; (c) 2020 NetApp, Inc.; Licensed to XXX [NetApp Inc] until Mon Dec 31 00:00:00 2029
753 scanned, 0 errors, 0 skipped, 0 copied, 0 (0/s), 5s
753 scanned, 0 errors, 0 skipped, 0 copied, 0 (0/s), 10s
753 scanned, 0 errors, 0 skipped, 0 copied, 0 (0/s), 15s
753 scanned, 0 errors, 0 skipped, 210 copied, 20.4KiB (4.08KiB/s), 20s
753 scanned, 0 errors, 0 skipped, 752 copied, 54.7KiB (2.38KiB/s), 22s
C:\WRSHDNT>

With the root folder, run the following commands:

C:\WRSHDNT>c:\netapp\xcp\xcp copy -acl -root -fallback-user \
D60AB15C2AFC4D6\Administrator -fallback-group BUILTIN\Users \
10.237.165.79\source_share 10.237.165.89\dest_share 
C:\netapp\xcp\xcp copy -acl -root -fallback-user \
D60AB15C2AFC4D6\Administrator -fallback-group BUILTIN\Users \
10.237.165.79\source_share 10.237.165.89\dest_share 
XCP SMB 1.6; (c) 2020 NetApp, Inc.; Licensed to XXX [NetApp Inc] until Mon Dec 31 00:00:00 2029
753 scanned, 0 errors, 0 skipped, 0 copied, 0 (0/s), 5s
753 scanned, 0 errors, 0 skipped, 0 copied, 0 (0/s), 10s
753 scanned, 0 errors, 0 skipped, 0 copied, 0 (0/s), 15s
753 scanned, 0 errors, 0 skipped, 243 copied, 23.6KiB (4.73KiB/s), 20s
753 scanned, 0 errors, 0 skipped, 752 copied, 54.7KiB (6.21KiB/s), 25s
753 scanned, 0 errors, 0 skipped, 752 copied, 54.7KiB (0/s), 30s
753 scanned, 0 errors, 0 skipped, 752 copied, 54.7KiB (0/s), 35s
753 scanned, 0 errors, 0 skipped, 752 copied, 54.7KiB (0/s), 40s
753 scanned, 0 errors, 0 skipped, 752 copied, 54.7KiB (0/s), 45s
753 scanned, 0 errors, 0 skipped, 752 copied, 54.7KiB (0/s), 50s
753 scanned, 0 errors, 0 skipped, 752 copied, 54.7KiB (0/s), 55s
753 scanned, 0 errors, 0 skipped, 752 copied, 54.7KiB (0/s), 1m0s
753 scanned, 0 errors, 0 skipped, 752 copied, 54.7KiB (0/s), 1m5s
753 scanned, 0 errors, 0 skipped, 752 copied, 54.7KiB (817/s), 1m8s
C:\WRSHDNT>

Best practice guidelines and recommendations

• Use the XCP client operating system, which is IMT supported. The IMT supported client is qualified by
NetApp.

- Run XCP as a root user in the Linux operating system to perform migration. You can run the xcp command as the sudo user, but it is not supported by XCP.
- Run only one instance of XCP per client. Technically, you can run multiple instances of XCP on the same host from a different location, however this is not a supported practice. Indeed, running many instances might result in failure.
- In the current XCP version, Live Source is not supported. If the source NetApp volume is active and continuously changed by applications and users, you should take a snapshot of the source volume to perform a migration.
- It is a best practice to create a new snapshot with a different name for every incremental sync so that it is easy to create an incremental migration path based on the snapshot name in the event of failure.
- If you are performing a snapshot-based migration, it is a best practice to continue snapshot-based migration until cutover.
- If you have more than 10 million files and you have incremental data change of more than 50%, it is a best practice to use a higher core count and more memory than the minimum recommendation in the installation and administration guide.

Troubleshooting

This section provides troubleshooting guidance for data migration using NetApp XCP.

Error 1: XCP Failed with nfs3 error 70: stale filehandle Error in the xcp.log

Reason and guidance.

Mount the source folder and verify that the folder exists. If it does not exist or if it has been removed, you will receive a stale filehandle error, in which case, you can ignore the error.

Error 2: NetApp NFS Destination Volume Has Space, but XCP Failed with nfs3 error 28: no space left on device

Reason and guidance.

1. Check the space of the NFS destination volume by running the df command or check the storage.

   ```bash
   root@workr-140: USER3# df -h /xcpsrc_vol
   Filesystem                 Size  Used Avail Use% Mounted on
   10.63.150.127:/xcpsrc_vol  4.3T  1.7T  2.6T  40% /xcpsrc_vol
   ```

2. Check the inodes in the storage controller.

   ```bash
   A800-Node1-2::> volume show -volume xcpdest -fields files,files-used
   vsserver       volume  files     files-used
   ------------------ -------- --------------------------
   A800-Node1_vs1  xcpdest  21251126 21251126
   A800-Node1-2::>
   ```

3. If inode is used, increase the number of inodes by running the following command:
A800-Node1-2::> volume modify -volume xcpdest -vserver A800-Node1_vs1-files 40000000
Volume modify successful on volume xcpdest of Vserver A800-Node1_vs1.
A800-Node1-2::> volume show -volume xcpdest -fields files,files-used
vserver  volume  files    files-used
-------------- ------- -------- ----------
A800-Node1_vs1 xcpdest 39999990 21251126
A800-Node1-2::>

Where to find additional information

To learn more about the information described in this document, refer to the following documents and/or websites:

• NetApp XCP blogs
  https://blog.netapp.com/tag/netapp-xcp/

• NetApp XCP user guide

• Bigdata Analytics data to Artificial Intelligence – Data mover solution for AI

Data Protection


Akash Gupta, NetApp
Girish Chanchlani, Commvault

TR-4320 outlines the reference architecture and best practices when using NetApp E-Series storage in a Commvault Data Platform V11 environment. Commvault and NetApp have jointly developed this reference architecture to provide guidance for Commvault Data Platform V11 deployments with NetApp E-Series storage that will accelerate time to application for this solution.


TR-4471: E-Series and EF-Series reference architecture and storage best practices with Veeam Backup & Replication 9.5

Akash Gupta, NetApp
Shawn Lieu (Americas), Stefan Renner (EMEA), and Michael Cade (Performance), Veeam

TR-4471 outlines the reference architecture and best practices when using NetApp E-Series storage in a Veeam Backup & Replication 9.5 environment.
TR-4704: Deploying Veritas NetBackup with NetApp E-Series storage

Akash Gupta and Principled Technologies, NetApp

TR-4704 describes deployment of Veritas NetBackup on NetApp E-Series storage.

NVA-1143: NetApp HCI - NIST security controls for FISMA with HyTrust for multitenant infrastructure - NVA design and deployment

Arvind Ramakrishnan, Abhinav Singh, NetApp

NVA-1143 describes how NetApp HCI can be designed and deployed to meet National Institute of Standards and Technology (NIST) SP 800-53 Revision 4 Security and Privacy controls, which are crucial for private cloud infrastructures and multitenant deployments.
Enterprise Applications
Purpose

Recovering a Very Large Database (VLDB) in Oracle using the Oracle Recovery Manager (RMAN) backup tool can be a highly challenging task. The database restoration process from backup media in the event of a failure can be time-consuming, delaying the database recovery and potentially impacting your Service Level Agreement (SLA) significantly. However, starting from version 10g, Oracle introduced a RMAN feature that allows users to create staged image copies of the Oracle database data files on additional disk storage located on the DB server host. These image copies can be incrementally updated using RMAN on a daily basis. In the case of a failure, the Database Administrator (DBA) can swiftly switch the Oracle database from the failed media to the image copy, eliminating the need for a complete database media restore. The result is a greatly improved SLA, albeit at the cost of doubling the required database storage.

If you are keen on SLA for your VLDB and contemplating moving the Oracle database to a public cloud such as Azure, you could set up a similar database protection structure using resources such as Microsoft Azure NetApp Files (ANF) for staging your standby database image copy. In this documentation, we demonstrate how to provision and export an NFS file system from ANF capacity pool to be mounted on an Oracle database server for staging a standby database copy for quick recovery in the event of a primary storage failure.

This solution addresses the following use cases:

• An Oracle VLDB image copy incremental merge via RMAN on NFS mount point off Microsoft ANF capacity pool storage.
• Quick recovery of an Oracle VLDB in the event of a failure on the same Azure database server VM.
• Quick recovery of an Oracle VLDB in the event of a failure on a standby Azure database server VM.

Audience

This solution is intended for the following people:

• A DBA who sets up Oracle VLDB image copy incremental merge via RMAN in Azure for faster database recovery.
• A database solution architect who tests Oracle workloads in the Azure public cloud.
• A storage administrator who manages Oracle databases deployed to ANF capacity pool storage.
• An application owner who would like to stand up Oracle databases in Azure cloud environment.

Solution test and validation environment

The testing and validation of this solution was performed in a Microsoft ANF capacity pool storage and Azure VM compute environment that might not match the final deployment environment. For more information, see the section [Key Factors for Deployment Consideration].
Oracle VLDB Incremental Merge via RMAN on ANF

**Hardware and software components**

**Hardware**

<table>
<thead>
<tr>
<th>Component</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANF storage</td>
<td>Current version offered by Microsoft, 2 TiB ANF capacity pool storage with Premium service level</td>
</tr>
<tr>
<td>Azure VM for DB server</td>
<td>Standard_B4ms - 4 vCPUs, 16GiB, 2 VMs, one as primary DB server and the other as a standby</td>
</tr>
</tbody>
</table>

**Software**

<table>
<thead>
<tr>
<th>Component</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>RedHat Linux</td>
<td>RHEL Linux 8.6 (LVM) - x64 Gen2, Deployed RedHat subscription for testing</td>
</tr>
<tr>
<td>Oracle Database</td>
<td>Version 19.18, Applied RU patch p34765931_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle OPatch</td>
<td>Version 12.2.0.1.36, Latest patch p6880880_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>NFS</td>
<td>Version 3.0, Oracle dNFS enabled</td>
</tr>
</tbody>
</table>

**Key factors for deployment consideration**

- **Oracle VLDB storage layout for RMAN incremental merge.** In our tests and validations, the NFS volume for Oracle incremental backup and merge is allocated from a single ANF capacity pool, which has...
100 TiB per volume, and 1000 TiB total capacity limit. For deployment over the thresholds, multiple volumes, and ANF capacity pools can be concatenated in parallel with multiple NFS mount points to provide higher capacity.

- **Oracle recoverability using RMAN incremental merge.** The RMAN incremental backup and merge is generally executed at user defined frequency based on your RTO and RPO objectives. If there are total loss of primary data storage and/or archived logs, the data loss can occur. The Oracle database can be recovered up to last incremental backup that is available from ANF database backup image copy. To minimize the data loss, Oracle flash recovery area can be setup on ANF NFS mount point and archived logs are backed up to ANF NFS mount along with database image copy.

- **Running Oracle VLDB off ANF NFS file system.** Unlike other bulk storage for database backup, Microsoft ANF is a cloud enabled production grade storage that delivers high level of performance and storage efficiency. Once Oracle VLDB switches over from primary storage to image copy on ANF NFS file system, database performance can be maintained at high level while the primary storage failure is addressed. You can take comfort to know that user application experience does not suffer as the result of primary storage failure.

- **Azure compute instances.** In these tests and validations, we used Standard_B4ms Azure VMs as the Oracle database servers. There are other Azure VMs that may be optimized and better suited for database workload. You also need to size the Azure VM appropriately for the number of vCPUs and the amount of RAM based on actual workload requirements.

- **ANF capacity pool service level.** ANF capacity pool offers three service level: Standard, Premium, Ultra. By default, an auto QoS applies to a volume created within a capacity pool, which restricts the throughput on the volume. The throughput on a volume can be manually adjusted based on the size of capacity pool and service level.

- **dNFS configuration.** dNFS is built into Oracle kernel and is known to dramatically increase Oracle database performance when Oracle is deployed to NFS storage. dNFS is packaged into Oracle binary but is not turned on by default. It should be turned on for any Oracle database deployment on NFS. For multiple ANF capacity pools deployment for a VLDB, dNFS multi-paths to different ANF capacity pools storage should be properly configured.

### Solution deployment

It is assumed that you already have your Oracle VLDB deployed in Azure cloud environment within a VNet. If you need help on Oracle deployment in Azure, please refer to following technical reports for help.

- **Simplified, Automated Oracle Deployment on Azure NetApp Files with NFS**
- **Oracle Database Deployment and Protection on Azure NetApp Files**

Your Oracle VLDB can be running either on an ANF storage or any other storage of choices within the Azure cloud ecosystem. The following section provides step-by-step deployment procedures for setting up RMAN incremental merge to an image copy of an Oracle VLDB that is staging in an NFS mount off ANF storage.

### Prerequisites for deployment
Deployment requires the following prerequisites.

1. An Azure account has been set up, and the necessary Azure VNet and network segments have been created within your Azure account.

2. From the Azure portal console, you must deploy two Azure VM instances, one as the primary Oracle DB server and an optional standby DB server. See the architecture diagram in the previous section for more details about the environment setup. Also review the Azure Virtual Machine series for more information.

3. From the Azure portal console, deploy ANF storage to host the NFS volumes that stores the Oracle database standby image copy. If you are not familiar with the deployment of ANF, see the documentation Quickstart: Set up Azure NetApp Files and create an NFS volume for step-by-step instructions.

Ensure that you have allocated at least 128G in Azure VM root volume in order to have sufficient space to stage Oracle installation files.

Provision and export NFS volume to be mounted on primary Oracle VLDB server
In this section, we show provisioning an NFS volume from an ANF capacity pool via Azure portal console. Repeat the procedures on other ANF capacity pools if more than one ANF capacity pools are set up to accommodate the size of the database.

1. First, from Azure portal console, navigating to ANF capacity pool that is used to stage Oracle VLDB image copy.

2. From selected capacity pool - database, click Volumes and then, Add volume to launch add-volume workflow.

3. Fill in Volume name, Quota, Virtual network, and Delegated subnet to move to Protocol page.
4. Take a note of the file path, enter allowed clients CIDR range, and enable Root Access for the volume.
5. Add a volume tag if desired.
6. Review and create the volume.
7. Login to primary Oracle VLDB server as a user with sudo privilege and mount the NFS volume exported from ANF storage. Change to your ANF NFS server IP address and file path as necessary. The ANF NFS server IP address can be retrieved from ANF volume console page.

```
sudo mkdir /nfsanf
```
sudo mount 172.30.136.68:/ora-01-u02-copy /nfsanf -o rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr

8. Change mount point ownership to oracle:oinstall, change to your oracle user name and primary group as necessary.

sudo chown oracle:oinstall /nfsanf

Setup Oracle RMAN incremental merge to image copy on ANF
RMAN incremental merge update the staging database data files image copy continuously at every incremental backup/merge interval. The image copy of database backup will be as up to date as the frequency you execute the incremental backup/merge. So, take into consideration of database performance, your RTO and RPO objectives when deciding the frequency of RMAN incremental backup and merge.

1. Login to primary Oracle VLDB server as oracle user.
2. Create an oracopy directory under mount point /nfsanf to store oracle data files image copies and archlog directory for Oracle flash recovery area.

```bash
mkdir /nfsanf/oracopy
mkdir /nfsanf/archlog
```

3. Login to Oracle database via sqlplus, enable block change tracking for faster incremental backup and change Oracle flash recovery area to ANF NFS mount if it is currently on primary storage. This allows the RMAN default control file/spfile autobackup and archived logs to be backed up to ANF NFS mount for recovery.

```bash
sqlplus / as sysdba
```

From sqlplus prompt, execute following command.

```sql
alter database enable block change tracking using file '/nfsanf/oracopy/bct_ntap1.ctf'
alter system set db_recovery_file_dest='/nfsanf/archlog/' scope=both;
```

Expected output:
4. Create a RMAN backup and incremental merge script. The script allocates multiple channels for parallel RMAN backup and merge. First execution would generate the initial full baseline image copy. In a complete run, it first purges obsolete backups that are outside of retention window to keep staging area clean. It then switches current log file before merge and backup. The incremental backup follows the merge so that the database image copy is trailing current database state by one backup/merge cycle. The merge and backup order can be reversed for quicker recovery at user’s preference. The RMAN script can be integrated into a simple shell script to be executed from crontab on the primary DB server. Ensure control file autobackup is on in RMAN setting.
vi /home/oracle/rman_bkup_merge.cmd

Add following lines:

RUN
{
    allocate channel c1 device type disk format '/nfsanf/oracopy/%U';
    allocate channel c2 device type disk format '/nfsanf/oracopy/%U';
    allocate channel c3 device type disk format '/nfsanf/oracopy/%U';
    allocate channel c4 device type disk format '/nfsanf/oracopy/%U';
    delete obsolete;
    sql 'alter system archive log current';
    recover copy of database with tag 'OraCopyBKUPonANF_level_0';
    backup incremental level 1 copies=1 for recover of copy with tag
    'OraCopyBKUPonANF_level_0' database;
}

5. At the primary Oracle VLDB server, login to RMAN locally as oracle user with or without RMAN catalog. In this demonstration, we are not connecting to a RMAN catalog.

```bash
rman target / nocatalog;
```

output:

```
[oracle@ora-01 ~]$ rman target / nocatalog

Recovery Manager: Release 19.0.0.0.0 - Production on Wed Mar 20 16:54:24 2024
Version 19.18.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.

carried out to target database: NTAP1 (DBID=2441823937)
using target database control file instead of recovery catalog
```

6. From RMAN prompt, execute the script. First execution creates a baseline database image copy and subsequent executions merge and update the baseline image copy incrementally. The following is how to execute the script and the typical output. Set the number of channels to match the CPU cores on the host.

```bash
RMAN> @/home/oracle/rman_bkup_merge.cmd
RMAN> RUN
```
allocate channel c1 device type disk format '/nfsanf/oracopy/%U';
allocate channel c2 device type disk format '/nfsanf/oracopy/%U';
allocate channel c3 device type disk format '/nfsanf/oracopy/%U';
allocate channel c4 device type disk format '/nfsanf/oracopy/%U';
delete obsolete;
sql 'alter system archive log current';
recover copy of database with tag 'OraCopyBKUPonANF_level_0';
backup incremental level 1 copies=1 for recover of copy with 
tag 'OraCopyBKUPonANF_level_0' database;

allocated channel: c1
channel c1: SID=142 device type=DISK
allocated channel: c2
channel c2: SID=277 device type=DISK
allocated channel: c3
channel c3: SID=414 device type=DISK
allocated channel: c4
channel c4: SID=28 device type=DISK

RMAN retention policy will be applied to the command
RMAN retention policy is set to redundancy 1
Deleting the following obsolete backups and copies:

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Completion Time</th>
<th>Filename/Handle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Set</td>
<td>1</td>
<td>18-MAR-24</td>
<td>/u03/orareco/NTAP1/autobackup/2024_03_18/o1 MF s_1163958359__04h19dg r_.bkp</td>
</tr>
<tr>
<td>Backup Set</td>
<td>2</td>
<td>18-MAR-24</td>
<td>/u03/orareco/NTAP1/autobackup/2024_03_18/o1 MF s_1163961675__07l1m2lg_.bkp</td>
</tr>
<tr>
<td>Backup Set</td>
<td>3</td>
<td>18-MAR-24</td>
<td>/u03/orareco/NTAP1/autobackup/2024_03_18/o1 MF s_1163961675__07l1m2lg_.bkp</td>
</tr>
<tr>
<td>Backup Set</td>
<td>4</td>
<td>18-MAR-24</td>
<td>/u03/orareco/NTAP1/autobackup/2024_03_18/o1 MF s_1163962888__08p6y7lx_.bkp</td>
</tr>
<tr>
<td>Backup Piece</td>
<td>1</td>
<td>18-MAR-24</td>
<td>/u03/orareco/NTAP1/autobackup/2024_03_18/o1 MF s_1163958359__04h19dg r_.bkp</td>
</tr>
<tr>
<td>Backup Piece</td>
<td>2</td>
<td>18-MAR-24</td>
<td>/u03/orareco/NTAP1/autobackup/2024_03_18/o1 MF s_1163961675__07l1m2lg_.bkp</td>
</tr>
<tr>
<td>Backup Piece</td>
<td>3</td>
<td>18-MAR-24</td>
<td>/u03/orareco/NTAP1/autobackup/2024_03_18/o1 MF s_1163961675__07l1m2lg_.bkp</td>
</tr>
<tr>
<td>Backup Piece</td>
<td>4</td>
<td>18-MAR-24</td>
<td>/u03/orareco/NTAP1/autobackup/2024_03_18/o1 MF s_1163962888__08p6y7lx_.bkp</td>
</tr>
</tbody>
</table>
deleted backup piece
backup piece
handle=/u03/orareco/NTAP1/autobackup/2024_03_18/o1_mf_s_1163961675__07l1m2lg_.bkp RECID=2 STAMP=1163961683
deleted backup piece
backup piece
handle=/u03/orareco/NTAP1/autobackup/2024_03_18/o1_mf_s_1163958359__04h19dgr_.bkp RECID=1 STAMP=1163958361
deleted backup piece
backup piece
handle=/u03/orareco/NTAP1/autobackup/2024_03_18/o1_mf_s_1163964697__0bd3tqg3_.bkp RECID=5 STAMP=1163964705
deleted backup piece
backup piece
handle=/u03/orareco/NTAP1/autobackup/2024_03_18/o1_mf_s_1163965895__0chx6mzt_.bkp RECID=6 STAMP=1163965906
deleted backup piece
backup piece
handle=/u03/orareco/NTAP1/autobackup/2024_03_18/o1_mf_s_1163966806__0dbyx344_.bkp RECID=7 STAMP=1163966814
deleted backup piece
backup piece
handle=/u03/orareco/NTAP1/autobackup/2024_03_18/o1_mf_s_1163968012__0fgvg805_.bkp RECID=8 STAMP=1163968018
deleted backup piece
backup piece
handle=/u03/orareco/NTAP1/autobackup/2024_03_18/o1_mf_s_1163968919__0g9x5t1v_.bkp RECID=9 STAMP=1163968926
deleted backup piece
backup piece
handle=/u03/orareco/NTAP1/autobackup/2024_03_18/o1_mf_s_1163969821__0h4rfdzj_.bkp RECID=10 STAMP=1163969827
Deleted 3 objects

deleted backup piece
backup piece
handle=/u03/orareco/NTAP1/autobackup/2024_03_18/o1_mf_s_1163971026__0j8o4wk8_.bkp RECID=11 STAMP=1163971032
Deleted 3 objects

deleted backup piece
backup piece
handle=/u03/orareco/NTAP1/autobackup/2024_03_18/o1_mf_s_1163971931__0k3pnn2o_.bkp RECID=12 STAMP=1163971938
Deleted 3 objects
Deleted backup piece
backup piece
handle=/u03/orareco/NTAP1/autobackup/2024_03_18/o1_mf_s_1163972835__0kyg92t1_.bkp RECID=13 STAMP=1163972837
Deleted 4 objects

sql statement: alter system archive log current

Starting recover at 20-MAR-24
no copy of datafile 1 found to recover
no copy of datafile 3 found to recover
no copy of datafile 4 found to recover
.
.
o copy of datafile 31 found to recover
no copy of datafile 32 found to recover
Finished recover at 20-MAR-24

Starting backup at 20-MAR-24
no parent backup or copy of datafile 1 found
no parent backup or copy of datafile 3 found
no parent backup or copy of datafile 4 found
.
.
o parent backup or copy of datafile 19 found
no parent backup or copy of datafile 20 found
channel c1: starting datafile copy
input datafile file number=00021
name=/u02/oradata/NTAP1/NTAP1_pdb1/soe_01.dbf
channel c2: starting datafile copy
input datafile file number=00022
name=/u02/oradata/NTAP1/NTAP1_pdb1/soe_02.dbf
channel c3: starting datafile copy
input datafile file number=00023
name=/u02/oradata/NTAP1/NTAP1_pdb1/soe_03.dbf
channel c4: starting datafile copy
input datafile file number=00024
name=/u02/oradata/NTAP1/NTAP1_pdb1/soe_04.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-22_0g2m6brl tag=ORACOPYBKUPONANF_LEVEL_0 RECID=4 STAMP=1164132108
channel c2: datafile copy complete, elapsed time: 01:06:39
channel c2: starting datafile copy
input datafile file number=00025
name=/u02/oradata/NTAP1/NTAP1_pdb1/soe_05.dbf
channel c4: datafile copy complete, elapsed time: 01:06:45
channel c4: starting datafile copy
input datafile file number=00026
name=/u02/oradata/NTAP1/NTAP1_pdb1/soe_06.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-24_0i2m6brl tag=ORACOPYBKUPONANF_LEVEL_0 RECID=5 STAMP=1164132121

channel c3: datafile copy complete, elapsed time: 01:08:05
channel c3: starting datafile copy
input datafile file number=00027
name=/u02/oradata/NTAP1/NTAP1_pdb1/soe_07.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-23_0h2m6brl tag=ORACOPYBKUPONANF_LEVEL_0 RECID=6 STAMP=1164132198

channel c1: datafile copy complete, elapsed time: 01:08:57
channel c1: starting datafile copy
input datafile file number=00028
name=/u02/oradata/NTAP1/NTAP1_pdb1/soe_08.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-25_0j2m6fol tag=ORACOPYBKUPONANF_LEVEL_0 RECID=7 STAMP=1164136123

channel c2: datafile copy complete, elapsed time: 01:06:46
channel c2: starting datafile copy
input datafile file number=00029
name=/u02/oradata/NTAP1/NTAP1_pdb1/soe_09.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-26_0k2m6fot tag=ORACOPYBKUPONANF_LEVEL_0 RECID=8 STAMP=1164136113

channel c4: datafile copy complete, elapsed time: 01:06:36
channel c4: starting datafile copy
input datafile file number=00030
name=/u02/oradata/NTAP1/NTAP1_pdb1/soe_10.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-27_0l2m6frc tag=ORACOPYBKUPONANF_LEVEL_0 RECID=9 STAMP=1164136293

channel c3: datafile copy complete, elapsed time: 01:08:10
channel c3: starting datafile copy
input datafile file number=00031
name=/u02/oradata/NTAP1/NTAP1_pdb1/soe_11.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-28_0m2m6fsu tag=ORACOPYBKUPONANF_LEVEL_0 RECID=10 STAMP=1164136333

channel c1: datafile copy complete, elapsed time: 01:07:52
channel c1: starting datafile copy
input datafile file number=00032
name=/u02/oradata/NTAP1/NTAP1_pdb1/soe_12.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SOE_FNO-29_0n2m6jlr tag=ORACOPYBKUPONANF_LEVEL_0 RECID=12
STAMP=1164140082
channel c2: datafile copy complete, elapsed time: 01:06:01
channel c2: starting datafile copy
input datafile file number=00001
name=/u02/oradata/NTAP1/system01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SOE_FNO-30_0o2m6jlr tag=ORACOPYBKUPONANF_LEVEL_0 RECID=13
STAMP=1164140190
channel c4: datafile copy complete, elapsed time: 01:07:49
channel c4: starting datafile copy
input datafile file number=00003
name=/u02/oradata/NTAP1/sysaux01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SYSTEM_FNO-1_0r2m6nhk tag=ORACOPYBKUPONANF_LEVEL_0 RECID=14
STAMP=1164140240
channel c2: datafile copy complete, elapsed time: 00:02:38
channel c2: starting datafile copy
input datafile file number=00004
name=/u02/oradata/NTAP1/undotbs01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
UNDOTBS1_FNO-4_0t2m6nml tag=ORACOPYBKUPONANF_LEVEL_0 RECID=15
STAMP=1164140372
channel c2: datafile copy complete, elapsed time: 00:02:15
channel c2: starting datafile copy
input datafile file number=00011
name=/u02/oradata/NTAP1/NTAP1_pdb1/undotbs01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SYSAUX_FNO-3_0s2m6nl1 tag=ORACOPYBKUPONANF_LEVEL_0 RECID=16
STAMP=1164140377
channel c4: datafile copy complete, elapsed time: 00:03:01
channel c4: starting datafile copy
input datafile file number=00010
name=/u02/oradata/NTAP1/NTAP1_pdb1/sysaux01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SOE_FNO-32_0q2m6jsi tag=ORACOPYBKUPONANF_LEVEL_0 RECID=17
STAMP=1164140385
channel c1: datafile copy complete, elapsed time: 01:07:29
channel c1: starting datafile copy
input datafile file number=00014
name=/u02/oradata/NTAP1/NTAP1_pdb2/sysaux01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SOE_FNO-33_1p2m6jml tag=ORACOPYBKUPONANF_LEVEL_0 RECID=18
STAMP=1164140449
channel c3: datafile copy complete, elapsed time: 01:08:31
input datafile file number=00018
name=/u02/oradata/NTAP1/NTAP1_pdb3/sysaux01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-10_0v2m6nqs tag=ORACOPYBKUPONANF_LEVEL_0 RECID=19
STAMP=1164140459

channel c4: datafile copy complete, elapsed time: 00:01:26
input datafile file number=00006
name=/u02/oradata/NTAP1/pdbseed/sysaux01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-14_102m6nr3 tag=ORACOPYBKUPONANF_LEVEL_0 RECID=20
STAMP=1164140468

channel c1: datafile copy complete, elapsed time: 00:01:22
input datafile file number=00009
name=/u02/oradata/NTAP1/NTAP1_pdb1/system01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-11_0u2m6nqs tag=ORACOPYBKUPONANF_LEVEL_0 RECID=21
STAMP=1164140471

channel c2: datafile copy complete, elapsed time: 00:01:33
input datafile file number=00013
name=/u02/oradata/NTAP1/NTAP1_pdb2/system01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-18_112m6nrt tag=ORACOPYBKUPONANF_LEVEL_0 RECID=22
STAMP=1164140476

channel c3: datafile copy complete, elapsed time: 00:00:57
input datafile file number=00017
name=/u02/oradata/NTAP1/NTAP1_pdb3/system01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-6_122m6nti tag=ORACOPYBKUPONANF_LEVEL_0 RECID=23
STAMP=1164140488

channel c4: datafile copy complete, elapsed time: 00:00:25
input datafile file number=00005
name=/u02/oradata/NTAP1/pdbseed/system01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-13_142m6ntp tag=ORACOPYBKUPONANF_LEVEL_0 RECID=24
STAMP=1164140532

channel c2: datafile copy complete, elapsed time: 00:01:06
channel c2: starting datafile copy
input datafile file number=00008
name=/u02/oradata/NTAP1/pdbseed/undotbs01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SYSTEM_FNO-17_152m6nts tag=ORACOPYBKUPONANF_LEVEL_0 RECID=25
STAMP=1164140539
channel c3: datafile copy complete, elapsed time: 00:01:03
channel c3: starting datafile copy
input datafile file number=00015
name=/u02/oradata/NTAP1/NTAP1_pdb2/undotbs01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SYSTEM_FNO-9_132m6ntm tag=ORACOPYBKUPONANF_LEVEL_0 RECID=26
STAMP=1164140541
channel c1: datafile copy complete, elapsed time: 00:01:13
channel c1: starting datafile copy
input datafile file number=00019
name=/u02/oradata/NTAP1/NTAP1_pdb3/undotbs01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SYSTEM_FNO-5_162m6nuc tag=ORACOPYBKUPONANF_LEVEL_0 RECID=27
STAMP=1164140541
channel c4: datafile copy complete, elapsed time: 00:00:41
channel c4: starting datafile copy
input datafile file number=00007 name=/u02/oradata/NTAP1/users01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
UNDOTBS1_FNO-8_172m6nvr tag=ORACOPYBKUPONANF_LEVEL_0 RECID=28
STAMP=1164140552
channel c2: datafile copy complete, elapsed time: 00:00:16
channel c2: starting datafile copy
input datafile file number=00012
name=/u02/oradata/NTAP1/NTAP1_pdb1/users01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
UNDOTBS1_FNO-15_182m6nvs tag=ORACOPYBKUPONANF_LEVEL_0 RECID=30
STAMP=1164140561
channel c3: datafile copy complete, elapsed time: 00:00:24
channel c3: starting datafile copy
input datafile file number=00016
name=/u02/oradata/NTAP1/NTAP1_pdb2/users01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
USERS_FNO-7_1a2m6001 tag=ORACOPYBKUPONANF_LEVEL_0 RECID=29
STAMP=1164140560
channel c4: datafile copy complete, elapsed time: 00:00:16
channel c4: starting datafile copy
input datafile file number=00020
name=/u02/oradata/NTAP1/NTAP1_pdb3/users01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
UNDOTBS1_FNO-19_192m6nvv tag=ORACOPYBKUPONANF_LEVEL_0 RECID=31
STAMP=1164140564
channel c1: datafile copy complete, elapsed time: 00:00:21
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
USERS_FNO-12_1b2m6o0e tag=ORACOPYBKUPONANF_LEVEL_0 RECID=32
STAMP=1164140564
channel c2: datafile copy complete, elapsed time: 00:00:02
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
USERS_FNO-16_1c2m6o0k tag=ORACOPYBKUPONANF_LEVEL_0 RECID=34
STAMP=1164140565
channel c3: datafile copy complete, elapsed time: 00:00:01
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
USERS_FNO-20_1d2m6o0k tag=ORACOPYBKUPONANF_LEVEL_0 RECID=33
STAMP=1164140565
channel c4: datafile copy complete, elapsed time: 00:00:01
Finished backup at 20-MAR-24

Starting Control File and SPFILE Autobackup at 20-MAR-24
piece
handle=/nfsanf/archlog/NTAP1/autobackup/2024_03_20/o1_mf_s_116414056
5__5g56y5ks_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 20-MAR-24
released channel: c1
released channel: c2
released channel: c3
released channel: c4

RMAN> **end-of-file**

RMAN> 7. List database image copy after backup to observe that a database image copy has been created in
ANF NFS mount point.

RMAN> list copy of database tag 'OraCopyBKUPonANF_level_0';

List of Datafile Copies
========================
<table>
<thead>
<tr>
<th>Key</th>
<th>File S</th>
<th>Completion Time</th>
<th>Ckp SCN</th>
<th>Ckp Time</th>
<th>Sparse</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>1</td>
<td>A 20-MAR-24</td>
<td>4161498</td>
<td>20-MAR-24</td>
<td>NO</td>
</tr>
</tbody>
</table>
|     | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SYSTEM_FNO-1_0r2m6nhk
<p>|     | Tag: ORACOPYBKUPONANF_LEVEL_0 |
| 16  | 3      | A 20-MAR-24     | 4161568 | 20-MAR-24| NO     |
|     | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- |</p>
<table>
<thead>
<tr>
<th>Tag: ORACOPYBKUPONANF_LEVEL_0</th>
</tr>
</thead>
<tbody>
<tr>
<td>15  4  A 20-MAR-24  4161589  20-MAR-24  NO</td>
</tr>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-UNDOTBS1_FNO-4_0t2m6nml</td>
</tr>
<tr>
<td>Tag: ORACOPYBKUPONANF_LEVEL_0</td>
</tr>
<tr>
<td>27  5  A 20-MAR-24  2379694  18-MAR-24  NO</td>
</tr>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-5_162m6nuc</td>
</tr>
<tr>
<td>Container ID: 2, PDB Name: PDB$SEED</td>
</tr>
<tr>
<td>23  6  A 20-MAR-24  2379694  18-MAR-24  NO</td>
</tr>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-6_122m6nti</td>
</tr>
<tr>
<td>Container ID: 2, PDB Name: PDB$SEED</td>
</tr>
<tr>
<td>29  7  A 20-MAR-24  4161872  20-MAR-24  NO</td>
</tr>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-USERS_FNO-7_1a2m6o01</td>
</tr>
<tr>
<td>Tag: ORACOPYBKUPONANF_LEVEL_0</td>
</tr>
<tr>
<td>28  8  A 20-MAR-24  2379694  18-MAR-24  NO</td>
</tr>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-UNDOTBS1_FNO-8_172m6nvr</td>
</tr>
<tr>
<td>Container ID: 2, PDB Name: PDB$SEED</td>
</tr>
<tr>
<td>26  9  A 20-MAR-24  4161835  20-MAR-24  NO</td>
</tr>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-9_132m6ntm</td>
</tr>
<tr>
<td>Container ID: 3, PDB Name: NTAP1_PDB1</td>
</tr>
<tr>
<td>19 10  A 20-MAR-24  4161784  20-MAR-24  NO</td>
</tr>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-10_0v2m6nqs</td>
</tr>
<tr>
<td>Container ID: 3, PDB Name: NTAP1_PDB1</td>
</tr>
<tr>
<td>21 11  A 20-MAR-24  4161780  20-MAR-24  NO</td>
</tr>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-UNDOTBS1_FNO-11_0u2m6nqs</td>
</tr>
<tr>
<td>Container ID</td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>34</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>31</td>
</tr>
</tbody>
</table>
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-UNDOTBS1_FNO-19_192m6nv
  Tag: ORACOPYBKUPONANF_LEVEL_0
  Container ID: 5, PDB Name: NTAP1_PDB3

33  20  A 20-MAR-24      4161887  20-MAR-24       NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-USERS_FNO-20_1d2m6o0k
  Tag: ORACOPYBKUPONANF_LEVEL_0
  Container ID: 5, PDB Name: NTAP1_PDB3

7   21  A 20-MAR-24      4152514  20-MAR-24       NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-21_0f2m6brl
  Tag: ORACOPYBKUPONANF_LEVEL_0
  Container ID: 3, PDB Name: NTAP1_PDB1

4   22  A 20-MAR-24      4152518  20-MAR-24       NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-22_0g2m6brl
  Tag: ORACOPYBKUPONANF_LEVEL_0
  Container ID: 3, PDB Name: NTAP1_PDB1

6   23  A 20-MAR-24      4152522  20-MAR-24       NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-23_0h2m6brl
  Tag: ORACOPYBKUPONANF_LEVEL_0
  Container ID: 3, PDB Name: NTAP1_PDB1

5   24  A 20-MAR-24      4152529  20-MAR-24       NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-24_0i2m6brl
  Tag: ORACOPYBKUPONANF_LEVEL_0
  Container ID: 3, PDB Name: NTAP1_PDB1

9   25  A 20-MAR-24      4156120  20-MAR-24       NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-25_0j2m6fol
  Tag: ORACOPYBKUPONANF_LEVEL_0
  Container ID: 3, PDB Name: NTAP1_PDB1

8   26  A 20-MAR-24      4156130  20-MAR-24       NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-26_0k2m6fot
  Tag: ORACOPYBKUPONANF_LEVEL_0
  Container ID: 3, PDB Name: NTAP1_PDB1
| No. | Date   | Time | Size (MB) | Date | Append | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-
|-----|--------|------|-----------|------|--------|---------------------------------------------------------
| 10  | 20-MAR-24 | 4156159 | 20-MAR-24 | NO   | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-
|     |         |       |           |      | Tag: ORACOPYBKUPONANF_LEVEL_0 | Container ID: 3, PDB Name: NTAP1_PDB1 |
| 11  | 20-MAR-24 | 4156183 | 20-MAR-24 | NO   | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-
|     |         |       |           |      | Tag: ORACOPYBKUPONANF_LEVEL_0 | Container ID: 3, PDB Name: NTAP1_PDB1 |
| 12  | 20-MAR-24 | 4158795 | 20-MAR-24 | NO   | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-
|     |         |       |           |      | Tag: ORACOPYBKUPONANF_LEVEL_0 | Container ID: 3, PDB Name: NTAP1_PDB1 |
| 13  | 20-MAR-24 | 4158803 | 20-MAR-24 | NO   | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-
|     |         |       |           |      | Tag: ORACOPYBKUPONANF_LEVEL_0 | Container ID: 3, PDB Name: NTAP1_PDB1 |
| 18  | 20-MAR-24 | 4158871 | 20-MAR-24 | NO   | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-
|     |         |       |           |      | Tag: ORACOPYBKUPONANF_LEVEL_0 | Container ID: 3, PDB Name: NTAP1_PDB1 |
| 17  | 20-MAR-24 | 4158886 | 20-MAR-24 | NO   | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-
|     |         |       |           |      | Tag: ORACOPYBKUPONANF_LEVEL_0 | Container ID: 3, PDB Name: NTAP1_PDB1 |

8. Report schema from Oracle RMAN command prompt to observe that current VLDB data files are on primary storage.

```
RMAN> report schema;
```

Report of database schema for database with db_unique_name NTAP1

List of Permanent Datafiles
===========================
```
<table>
<thead>
<tr>
<th>File Size(MB)</th>
<th>Tablespace</th>
<th>RB segs</th>
<th>Datafile Name</th>
</tr>
</thead>
</table>
```

715
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1060</td>
<td>SYSTEM</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/system01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>SYSAUX</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/sysaux01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>695</td>
<td>UNDOTBS1</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/undotbs01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>400</td>
<td>PDB$SEED:SYSTEM</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/pdbseed/system01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>440</td>
<td>PDB$SEED:SYSAUX</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/pdbseed/sysaux01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>USERS</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/users01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>235</td>
<td>PDB$SEED:UNDOTBS1</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/pdbseed/undotbs01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>410</td>
<td>NTAP1_PDB1:SYSTEM</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/NTAP1_pdb1/system01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>520</td>
<td>NTAP1_PDB1:SYSAUX</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/NTAP1_pdb1/sysaux01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>580</td>
<td>NTAP1_PDB1:UNDOTBS1</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/NTAP1_pdb1/undotbs01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>NTAP1_PDB1:USERS</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/NTAP1_pdb1/users01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>410</td>
<td>NTAP1_PDB2:SYSTEM</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/NTAP1_pdb2/system01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>500</td>
<td>NTAP1_PDB2:SYSAUX</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/NTAP1_pdb2/sysaux01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>235</td>
<td>NTAP1_PDB2:UNDOTBS1</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/NTAP1_pdb2/undotbs01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>5</td>
<td>NTAP1_PDB2:USERS</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/NTAP1_pdb2/users01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>410</td>
<td>NTAP1_PDB3:SYSTEM</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/NTAP1_pdb3/system01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>500</td>
<td>NTAP1_PDB3:SYSAUX</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/NTAP1_pdb3/sysaux01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>235</td>
<td>NTAP1_PDB3:UNDOTBS1</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/NTAP1_pdb3/undotbs01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>NTAP1_PDB3:USERS</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/NTAP1_pdb3/users01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>31744</td>
<td>NTAP1_PDB1:SOE</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/NTAP1_pdb1/soe_01.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>31744</td>
<td>NTAP1_PDB1:SOE</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/NTAP1_pdb1/soe_02.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>31744</td>
<td>NTAP1_PDB1:SOE</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>/u02/oradata/NTAP1/NTAP1_pdb1/soe_03.dbf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>31744</td>
<td>NTAP1_PDB1:SOE</td>
<td>NO</td>
</tr>
</tbody>
</table>
List of Temporary Files
================================

<table>
<thead>
<tr>
<th>File Size(MB)</th>
<th>Tablespace</th>
<th>Maxsize(MB)</th>
<th>Tempfile Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>----</td>
<td>--------</td>
<td>--------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>1</td>
<td>123</td>
<td>TEMP</td>
<td>32767</td>
</tr>
<tr>
<td>/u02/oradata/NTAP1/temp01.dbf</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>123</td>
<td>PDB$SEED:TEMP</td>
<td>32767</td>
</tr>
<tr>
<td>/u02/oradata/NTAP1/pdbseed/temp012024-03-18_16-07-32-463-PM.dbf</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>31744</td>
<td>NTAP1_PDB1:TEMP</td>
<td>32767</td>
</tr>
<tr>
<td>/u02/oradata/NTAP1/NTAP1_pdb1/temp01.dbf</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>123</td>
<td>NTAP1_PDB2:TEMP</td>
<td>32767</td>
</tr>
<tr>
<td>/u02/oradata/NTAP1/NTAP1_pdb2/temp01.dbf</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>123</td>
<td>NTAP1_PDB3:TEMP</td>
<td>32767</td>
</tr>
<tr>
<td>/u02/oradata/NTAP1/NTAP1_pdb3/temp01.dbf</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>31744</td>
<td>NTAP1_PDB1:TEMP</td>
<td>31744</td>
</tr>
<tr>
<td>/u02/oradata/NTAP1/NTAP1_pdb1/temp02.dbf</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RMAN>

9. Validate database image copy from OS NFS mount point.

[oracle@ora-01 ~]$ ls -l /nfsanf/oracopy
total 399482176
-rw-r----- 1 oracle oinstall 11600384 Mar 20 21:44 bct_ntap1.ctf
-rw-r----- 1 oracle oinstall 33286004736 Mar 20 18:03 data_D-NTAP1_I-2441823937_TS-SOE_FNO-21_0f2m6brl
-rw-r----- 1 oracle oinstall 33286004736 Mar 20 18:01 data_D-
This completes the setup of an Oracle VLDB standby image copy backup and merge.

Switch Oracle VLDB to image copy for quick recovery
In the event of a failure due to primary storage issue such as data loss or corruption, database can be quickly switched over to image copy on ANF NFS mount and recovered to current state without database restore. Eliminating media restoration speeds up the database recovery tremendously for a VLDB. This use case assumes that the Oracle VLDB DB server is intact and database control file, archived and current logs are all available for recovery.

1. Login to Azure primary VLDB server host as oracle user and create a test table before switch over.

```
[oracle@ora-01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Mar 21 15:13:52 2024
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.18.0.0.0

SQL> show pdbs

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>CON_NAME</th>
<th>OPEN_MODE</th>
<th>RESTRICTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>READ ONLY</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>NTAP1_PDB1</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>4</td>
<td>NTAP1_PDB2</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>5</td>
<td>NTAP1_PDB3</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
</tbody>
</table>

SQL> alter session set container=ntap1_pdb1;

Session altered.

SQL> create table test (id integer, dt timestamp, event varchar(100));

Table created.

SQL> insert into test values(1, sysdate, 'test oracle incremental merge switch to copy');

1 row created.

SQL> commit;
```
Commit complete.

SQL> select * from test;

<table>
<thead>
<tr>
<th>ID</th>
<th>DT</th>
<th>EVENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21-MAR-24 03.15.03.000000 PM</td>
<td>test oracle incremental merge switch to copy</td>
</tr>
</tbody>
</table>

2. Simulate a failure by shutdown abort database, then start up oracle in mount stage.

SQL> shutdown abort;  
ORACLE instance shut down.  
SQL> startup mount;  
ORACLE instance started.  

Total System Global Area 6442449688 bytes  
Fixed Size 9177880 bytes  
Variable Size 1325400064 bytes  
Database Buffers 5100273664 bytes  
Redo Buffers 7598080 bytes  
Database mounted.  
SQL> exit

3. As oracle user, connect to Oracle database via RMAN to switch database to copy.

[oracle@ora-01 ~]$ rman target / nocatalog

Recovery Manager: Release 19.0.0.0.0 - Production on Thu Mar 21 15:20:58 2024  
Version 19.18.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.

class connected to target database: NTAP1 (DBID=2441823937, not open)  
using target database control file instead of recovery catalog
RMAN> switch database to copy;

datafile 1 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSTEM_FNO-1_0r2m6nhk"
datafile 3 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSAUX_FNO-3_0s2m6n11"
datafile 4 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-UNDOTBS1_FNO-4_0t2m6nml"
datafile 5 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSTEM_FNO-5_162m6nuc"
datafile 6 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSAUX_FNO-6_122m6nti"
datafile 7 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-UNDOTBS1_FNO-8_172m6nvr"
datafile 8 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-USER_FNO-7_1a2m6o01"
datafile 9 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-UNDOTBS1_FNO-10_0v2m6nqs"
datafile 10 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-USERS_FNO-11_0u2m6nqs"
datafile 11 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-USERS_FNO-12_1b2m6o0e"
datafile 12 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSTEM_FNO-13_142m6ntp"
datafile 13 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-USER_FNO-15_182m6nvs"
datafile 14 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-USER_FNO-16_1c2m6o0k"
datafile 15 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSTEM_FNO-17_152m6nts"
datafile 16 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSTEM_FNO-18_112m6nrt"
datafile 17 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-UNDOTBS1_FNO-19_192m6nvv"
datafile 18 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-UNDOTBS1_FNO-20_1d2m6o0k"
datafile 19 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SOE_FNO-21_0f2m6brl"
datafile 20 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SOE_FNO-22_0g2m6brl"
datafile 21 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SOE_FNO-23_0h2m6brl"
datafile 24 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-24_0i2m6b1l"
datafile 25 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-25_0j2m6fol"
datafile 26 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-26_0k2m6fot"
datafile 27 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-27_0l2m6frc"
datafile 28 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-28_0m2m6fsu"
datafile 29 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-29_0n2m6jlr"
datafile 30 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-30_0o2m6jlr"
datafile 31 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-31_0p2m6jrb"
datafile 32 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-32_0q2m6jsi"

4. Recover and open database to bring it up to current from last incremental backup.

RMAN> recover database;

Starting recover at 21-MAR-24
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=392 device type=DISK
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
destination for restore of datafile 00009: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-9_0q1sd7c
destination for restore of datafile 00023: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-23_041sd6s5
destination for restore of datafile 00027: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-27_081sd70i
destination for restore of datafile 00031: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-31_0c1sd74u
destination for restore of datafile 00034: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-34_0f1sd788
channel ORA_DISK_1: reading from backup piece
/nfsanf/oracopy/321sfous_98_1_1
channel ORA_DISK_1: piece handle=/nfsanf/oracopy/321sfous_98_1_1
tag=ORACOPYBKUPONANF_LEVEL_0
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
destination for restore of datafile 00010: /nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSAUX_FNO-10_0k1sd7bb
destination for restore of datafile 00021: /nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SOE_FNO-21_021sd6pv
destination for restore of datafile 00025: /nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SOE_FNO-25_061sd6uc
.
.
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
destination for restore of datafile 00016: /nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-USERS_FNO-16_121sd7dn
channel ORA_DISK_1: reading from backup piece
/nfsanf/oracopy/3i1sfov0_114_1_1
channel ORA_DISK_1: piece handle=/nfsanf/oracopy/3i1sfov0_114_1_1
tag=ORACOPYBKUPONANF_LEVEL_0
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
destination for restore of datafile 00020: /nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-USERS_FNO-20_131sd7do
channel ORA_DISK_1: reading from backup piece
/nfsanf/oracopy/3j1sfov0_115_1_1
channel ORA_DISK_1: piece handle=/nfsanf/oracopy/3j1sfov0_115_1_1
tag=ORACOPYBKUPONANF_LEVEL_0
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01

starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 21-MAR-24

RMAN> alter database open;
Statement processed
RMAN>
5. Check database structure from sqlplus after recovery to observe that all VLDB data files with exception of control, temp, and current log files are now switched over to copy on ANF NFS file system.

```sql
SQL> select name from v$datafile
  2  union
  3  select name from v$tempfile
  4  union
  5  select name from v$controlfile
  6  union
  7* select member from v$logfile
SQL> /
```

<table>
<thead>
<tr>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-21_0f2m6brl</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-22_0g2m6brl</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-23_0h2m6brl</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-24_0i2m6brl</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-25_0j2m6fol</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-26_0k2m6fot</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-27_0l2m6frc</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-28_0m2m6fsu</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-29_0n2m6jlr</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-30_0o2m6jrb</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-31_0p2m6jsi</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-32_0q2m6jst</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-10_0v2m6nqs</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-14_02m6nr3</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-18_12m6nrt</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-3_132m6n11</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-6_112m6nti</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-13_142m6ntp</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-17_152m6nts</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-1_0r2m6nhk</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-5_162m6nuc</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-9_132m6ntm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-13_142m6ntp</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-17_152m6nts</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-1_0r2m6nhk</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-5_162m6nuc</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-9_132m6ntm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-13_142m6ntp</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-17_152m6nts</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-1_0r2m6nhk</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-5_162m6nuc</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-9_132m6ntm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-13_142m6ntp</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-17_152m6nts</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-1_0r2m6nhk</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-5_162m6nuc</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-9_132m6ntm</td>
</tr>
</tbody>
</table>
6. From SQL plus, check the content of test table we have inserted before the switch over to copy.
SQL> alter session set container=ntap1_pdb1;
Session altered.

SQL> select * from test;

ID
-----
DT
-----------------------------
-------
EVENT
-----------------------------
-------
1
21-MAR-24 03.15.03.000000 PM
test oracle incremental merge switch to copy

7. You could run the Oracle VLDB in ANF NFS mount for an extended period of time while maintaining expected performance level. When the primary storage issue is fixed, you can swing back to it by reversing the incremental backup merge processes with minimal downtime.

Oracle VLDB recovery from image copy to a standby DB server
In the event of a failure where both the primary storage and primary DB server host are lost, recovery cannot be performed from the original server. However, your Oracle database backup image copy available on the ANF NFS file system comes in handy. You can quickly recover the primary database to a standby DB server if one is available, using the backup image copy. In this section, we will demonstrate the step-by-step procedures for such recovery.

1. Insert a row to test table we have created previously for Oracle VLDB restoring to alternative host validation.
2. As oracle user, run RMAN incremental backup and merge to flush the transaction to backup set on ANF NFS mount.
3. Shutdown primary VLDB server host to simulate a total failure of storage and DB server host.

4. On the standby DB server ora-02 with same OS and version, OS kernel should be patched up as primary VLDB server host. Also, the same version and patches of Oracle has been installed and configured on standby DB server with software only option.

5. Configure oracle environment similarly to primary VLDB server ora_01, such as oratab, and oracle user .bash_profile etc. It is a good practice to backup those files to ANF NFS mount point.

6. The Oracle database backup image copy on ANF NFS file system is then mounted on the standby DB server for recovery. The following procedures demonstrate the process details.

As azureuser, create the mount point.

```bash
sudo mkdir /nfsanf
```

As azureuser, mount the NFS volume that stored Oracle VLDB backup image copy.

```bash
sudo mount 172.30.136.68:/ora-01-u02-copy /nfsanf -o rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr
```

7. Validate the Oracle database backup image copy on ANF NFS mount point.

```bash
[oracle@ora-02 ~]$ ls -ltr /nfsanf/oracopy/
total 400452728
-rw-r------. 1 oracle oinstall 461381632 Mar 21 23:47 data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-6_242m9oan
-rw-r------. 1 oracle oinstall 419438592 Mar 21 23:49 data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-5_282m9oem
-rw-r------. 1 oracle oinstall 246423552 Mar 21 23:49 data_D-NTAP1_I-2441823937_TS-UNDOTBS1_FNO-8_292m9oem
```
8. Verify the available Oracle archived logs on the ANF NFS mount for recovery and note the last log file log sequency number. In this case, it is 10. Our recovery point is up to log sequency number 11.
9. As oracle user, set ORACLE_HOME variable to current Oracle installation on standby DB server ora-02, ORACLE_SID to primary Oracle instance SID. In this case, it is NTAP1.

```
[oracle@ora-02 ~]$ export ORACLE_HOME=/u01/app/oracle/product/19.0.0/NTAP2
[oracle@ora-02 ~]$ export ORACLE_SID=NTAP1
[oracle@ora-02 ~]$ export PATH=$PATH:$ORACLE_HOME/bin
```

10. As oracle user, create a generic Oracle init file in $ORACLE_HOME/dbs directory with proper admin directories configured. Most importantly, have Oracle flash recovery area point to ANF NFS mount path as defined in primary Oracle VLDB server. flash recovery area configuration is demonstrated in section Setup Oracle RMAN incremental merge to image copy on ANF. Set the Oracle control file to ANF NFS file system.

```
vi $ORACLE_HOME/dbs/initNTAP1.ora
```

With following example entries:
The above init file should be replaced by restored backup init file from primary Oracle VLDB server in the case of discrepancy.

11. As oracle user, launch RMAN to run Oracle recovery on the standby DB server host. First, start the Oracle instance in nomount state.
12. Set database ID. The database ID can be retrieved from Oracle file name of image copy on ANF NFS mount point.

RMAN> set dbid = 2441823937;

executing command: SET DBID

13. Restore controlfile from autobackup. If Oracle controlfile and spfile autobackup is enabled, they are backed up in every incremental backup and merge cycle. The latest backup will be restored if multiple copies are available.
14. Restore init file from spfile to a /tmp folder for updating parameter file later to match with primary VLDB.

```
RMAN> restore spfile to pfile '/tmp/archive/initNTAP1.ora' from autobackup;

Starting restore at 22-MAR-24
using channel ORA_DISK_1

recovery area destination: /nfsanf/archlog/
database name (or database unique name) used for search: NTAP1
cchannel ORA_DISK_1: AUTOBACKUP
/nfsanf/archlog/NTAP1/autobackup/2024_03_22/o1_mf_s_1164296325__9z77zyxb_.b kp found in the recovery area
channel ORA_DISK_1: looking for AUTOBACKUP on day: 20240322
channel ORA_DISK_1: restoring spfile from AUTOBACKUP
/nfsanf/archlog/NTAP1/autobackup/2024_03_22/o1_mf_s_1164296325__9z77zyxb_.b kp
channel ORA_DISK_1: SPFILE restore from AUTOBACKUP complete
Finished restore at 22-MAR-24
```

15. Mount control file and validate the database backup image copy.

```
RMAN> alter database mount;
```
RMAN> list copy of database tag 'ORACOPYBKUPONANF_LEVEL_0';

List of Datafile Copies
=======================

<table>
<thead>
<tr>
<th>Key</th>
<th>File S</th>
<th>Completion Time</th>
<th>Ckp SCN</th>
<th>Ckp Time</th>
<th>Sparse</th>
<th>Name:</th>
<th>Tag:</th>
</tr>
</thead>
<tbody>
<tr>
<td>82</td>
<td>1</td>
<td>A 22-MAR-24</td>
<td>4598427</td>
<td>22-MAR-24</td>
<td>NO</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-1_1t2m9nij</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
</tr>
<tr>
<td>83</td>
<td>3</td>
<td>A 22-MAR-24</td>
<td>4598423</td>
<td>22-MAR-24</td>
<td>NO</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-3_1u2m9nog</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
</tr>
<tr>
<td>84</td>
<td>4</td>
<td>A 22-MAR-24</td>
<td>4598431</td>
<td>22-MAR-24</td>
<td>NO</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-UNDOTBS1_FNO-4_1v2m9nu6</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
</tr>
<tr>
<td>58</td>
<td>5</td>
<td>A 21-MAR-24</td>
<td>2379694</td>
<td>18-MAR-24</td>
<td>NO</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-5_282m9oem</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
</tr>
<tr>
<td>52</td>
<td>6</td>
<td>A 21-MAR-24</td>
<td>2379694</td>
<td>18-MAR-24</td>
<td>NO</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-6_242m9oan</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
</tr>
<tr>
<td>90</td>
<td>7</td>
<td>A 22-MAR-24</td>
<td>4598462</td>
<td>22-MAR-24</td>
<td>NO</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-USERS_FNO-7_2c2m9oan</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
</tr>
<tr>
<td>59</td>
<td>8</td>
<td>A 21-MAR-24</td>
<td>2379694</td>
<td>18-MAR-24</td>
<td>NO</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-UNDOTBS1_FNO-8_292m9oem</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
</tr>
</tbody>
</table>
Container ID: 2, PDB Name: PDB$SEED

71 9 A 22-MAR-24 4598313 22-MAR-24 NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SYSTEM_FNO-9_252m9oc5
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 3, PDB Name: NTAP1_PDB1

68 10 A 22-MAR-24 4598308 22-MAR-24 NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SYSAUX_FNO-10_212m9o52
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 3, PDB Name: NTAP1_PDB1

66 11 A 22-MAR-24 4598304 22-MAR-24 NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
UNDOTBS1_FNO-11_202m9o22
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 3, PDB Name: NTAP1_PDB1

74 12 A 22-MAR-24 4598318 22-MAR-24 NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
USERS_FNO-12_2d2m9ofs
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 3, PDB Name: NTAP1_PDB1

86 13 A 22-MAR-24 4598445 22-MAR-24 NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SYSTEM_FNO-13_262m9oca
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 4, PDB Name: NTAP1_PDB2

85 14 A 22-MAR-24 4598437 22-MAR-24 NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SYSAUX_FNO-14_222m9o53
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 4, PDB Name: NTAP1_PDB2

87 15 A 22-MAR-24 4598454 22-MAR-24 NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
UNDOTBS1_FNO-15_2a2m9of6
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 4, PDB Name: NTAP1_PDB2

89 16 A 22-MAR-24 4598466 22-MAR-24 NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
USERS_FNO-16_2e2m9og8
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 4, PDB Name: NTAP1_PDB2

91  17   A 22-MAR-24       4598450    22-MAR-24       NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-17_272m9oe1
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 5, PDB Name: NTAP1_PDB3

88  18   A 22-MAR-24       4598441    22-MAR-24       NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSaux_FNO-18_232m9oa8
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 5, PDB Name: NTAP1_PDB3

92  19   A 22-MAR-24       4598458    22-MAR-24       NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-UNDOTBS1_FNO-19_2b2m9ofn
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 5, PDB Name: NTAP1_PDB3

93  20   A 22-MAR-24       4598470    22-MAR-24       NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-USERS_FNO-20_2f2m9og8
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 5, PDB Name: NTAP1_PDB3

81  21   A 22-MAR-24       4598318    22-MAR-24       NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-21_1h2m9cap
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 3, PDB Name: NTAP1_PDB1

72  22   A 22-MAR-24       4598304    22-MAR-24       NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-22_1i2m9cap
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 3, PDB Name: NTAP1_PDB1

73  23   A 22-MAR-24       4598308    22-MAR-24       NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-23_1j2m9cap
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 3, PDB Name: NTAP1_PDB1
<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>24</td>
<td>22-MAR-24</td>
<td>4598313</td>
<td>22-MAR-24</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-24_1k2m9cap</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tag: ORACOPYBKUPONANF_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Container ID: 3, PDB Name: NTAP1_PDB1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| 79 | 25 | 22-MAR-24 | 4598318 | 22-MAR-24 | NO |
|    |   | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-25_1l2m9g3u |
|    |   | Tag: ORACOPYBKUPONANF_LEVEL_0 |
|    |   | Container ID: 3, PDB Name: NTAP1_PDB1 |

| 69 | 26 | 22-MAR-24 | 4598304 | 22-MAR-24 | NO |
|    |   | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-26_1m2m9g9j |
|    |   | Tag: ORACOPYBKUPONANF_LEVEL_0 |
|    |   | Container ID: 3, PDB Name: NTAP1_PDB1 |

| 70 | 27 | 22-MAR-24 | 4598308 | 22-MAR-24 | NO |
|    |   | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-27_1n2m9gcg |
|    |   | Tag: ORACOPYBKUPONANF_LEVEL_0 |
|    |   | Container ID: 3, PDB Name: NTAP1_PDB1 |

| 75 | 28 | 22-MAR-24 | 4598313 | 22-MAR-24 | NO |
|    |   | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-28_1o2m9gd4 |
|    |   | Tag: ORACOPYBKUPONANF_LEVEL_0 |
|    |   | Container ID: 3, PDB Name: NTAP1_PDB1 |

| 77 | 29 | 22-MAR-24 | 4598318 | 22-MAR-24 | NO |
|    |   | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-29_1p2m9ju6 |
|    |   | Tag: ORACOPYBKUPONANF_LEVEL_0 |
|    |   | Container ID: 3, PDB Name: NTAP1_PDB1 |

| 67 | 30 | 22-MAR-24 | 4598304 | 22-MAR-24 | NO |
|    |   | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-30_1q2m9k7a |
|    |   | Tag: ORACOPYBKUPONANF_LEVEL_0 |
|    |   | Container ID: 3, PDB Name: NTAP1_PDB1 |

| 76 | 31 | 22-MAR-24 | 4598308 | 22-MAR-24 | NO |
|    |   | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-31_1r2m9kfk |
|    |   | Tag: ORACOPYBKUPONANF_LEVEL_0 |
16. Switch database to copy to run recovery without database restore.

RMAN> switch database to copy;

Starting implicit crosscheck backup at 22-MAR-24
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=12 device type=DISK
Crosschecked 33 objects
Finished implicit crosscheck backup at 22-MAR-24

Starting implicit crosscheck copy at 22-MAR-24
using channel ORA_DISK_1
Crosschecked 31 objects
Finished implicit crosscheck copy at 22-MAR-24

searching for all files in the recovery area
cataloging files...
cataloging done

List of Cataloged Files
=======================
File Name:
/nfsanf/archlog/NTAP1/autobackup/2024_03_20/o1_mf_s_1164140565__5g56ypks_.bkp
File Name:
/nfsanf/archlog/NTAP1/autobackup/2024_03_22/o1_mf_s_1164296325__9z77zyxb_.bkp

datafile 1 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSTEM_FNO-1_1t2m9nij"
datafile 3 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSAUX_FNO-3_1u2m9nog"
datafile 4 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-UNDOTBS1_FNO-4_1v2m9nu6"
datafile 5 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSTEM_FNO-5_282m9oem"
datafile 6 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSAUX_FNO-6_242m9oan"
datafile 7 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- USERS_FNO-7_2c2m9ofn"
datafile 8 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- UNDOTBS1_FNO-8_292m9oem"
datafile 9 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- SYSTEM_FNO-9_252m9oc5"
datafile 10 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- SYSAUX_FNO-10_212m9o52"
datafile 11 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- UNDOTBS1_FNO-11_202m9o22"
datafile 12 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- USERS_FNO-12_2d2m9ofs"
datafile 13 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- SYSTEM_FNO-13_262m9oca"
datafile 14 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- SYSAUX_FNO-14_222m9o53"
datafile 15 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- UNDOTBS1_FNO-15_2a2m9of6"
datafile 16 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- USERS_FNO-16_2e2m9og8"
datafile 17 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- SYSTEM_FNO-17_272m9oel"
datafile 18 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- SYSAUX_FNO-18_232m9oa8"
datafile 19 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- UNDOTBS1_FNO-19_2b2m9ofn"
datafile 20 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- USERS_FNO-20_2f2m9og8"
datafile 21 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- SOE_FNO-21_1h2m9cap"
datafile 22 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- SOE_FNO-22_1i2m9cap"
datafile 23 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- SOE_FNO-23_1j2m9cap"
datafile 24 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- SOE_FNO-24_1k2m9cap"
datafile 25 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- SOE_FNO-25_1l2m9g3u"
datafile 26 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- SOE_FNO-26_1m2m9g9j"
datafile 27 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- SOE_FNO-27_1n2m9gcg"
datafile 28 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- SOE_FNO-28_1o2m9gd4"
datafile 29 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- SOE_FNO-29_1p2m9ju6"
17. Run Oracle recovery up to last available archive log in flash recovery area.

RMAN> run {
  2> set until sequence=11;
  3> recover database;
  4> }

executing command: SET until clause

Starting recover at 22-MAR-24
using channel ORA_DISK_1

starting media recovery

archived log for thread 1 with sequence 4 is already on disk as file
/nfsanf/archlog/NTAP1/archivelog/2024_03_22/o1_mf_1_4__9y6gn5co_.arc
archived log for thread 1 with sequence 5 is already on disk as file
/nfsanf/archlog/NTAP1/archivelog/2024_03_22/o1_mf_1_5__9y7p68s6_.arc
archived log for thread 1 with sequence 6 is already on disk as file
/nfsanf/archlog/NTAP1/archivelog/2024_03_22/o1_mf_1_6__9y8ygtss_.arc
archived log for thread 1 with sequence 7 is already on disk as file
/nfsanf/archlog/NTAP1/archivelog/2024_03_22/o1_mf_1_7__9ybjdp55_.arc
archived log for thread 1 with sequence 8 is already on disk as file
/nfsanf/archlog/NTAP1/archivelog/2024_03_22/o1_mf_1_8__9yctxjgy_.arc
archived log for thread 1 with sequence 9 is already on disk as file
/nfsanf/archlog/NTAP1/archivelog/2024_03_22/o1_mf_1_9__9yfrj0bl_.arc
archived log for thread 1 with sequence 10 is already on disk as file
/nfsanf/archlog/NTAP1/archivelog/2024_03_22/o1_mf_1_10__9yomybbc_.arc
archived log file
name=/nfsanf/archlog/NTAP1/archivelog/2024_03_22/o1_mf_1_4__9y6gn5co___.arc thread=1 sequence=4
archived log file
name=/nfsanf/archlog/NTAP1/archivelog/2024_03_22/o1_mf_1_5__9y7p68s6___.arc thread=1 sequence=5
archived log file
name=/nfsanf/archlog/NTAP1/archivelog/2024_03_22/o1_mf_1_6__9y8ygtss
For faster recovery, enable parallel sessions with recovery_parallelism parameter or specify degree of parallel in recovery command for database recovery: RECOVER DATABASE PARALLEL (DEGREE d INSTANCES DEFAULT); In general, degrees of parallelism should be equal to number of CPU cores on the host.

18. Exit RMAN, login to Oracle as oracle user via sqlplus to open database and reset log after an incomplete recovery.

    SQL> select name, open_mode from v$database;

    NAME        OPEN_MODE
    ---------- ---------------
    NTAP1       MOUNTED

    SQL> select instance_name, host_name from v$instance;

    INSTANCE_NAME
    -------------
    NTAP1
   
    HOST_NAME
    -------------
    NTAP1
    ora-02

    SQL>
SQL> select member from v$logfile;

MEMBER
--------------------------------------------------------------------
-----------
/u03/orareco/NTAP1/onlinelog/redo03.log
/u03/orareco/NTAP1/onlinelog/redo02.log
/u03/orareco/NTAP1/onlinelog/redo01.log

SQL> alter database rename file
'/u03/orareco/NTAP1/onlinelog/redo01.log' to
'/nfsanf/oracopy/redo01.log';

Database altered.

SQL> alter database rename file
'/u03/orareco/NTAP1/onlinelog/redo02.log' to
'/nfsanf/oracopy/redo02.log';

Database altered.

SQL> alter database rename file
'/u03/orareco/NTAP1/onlinelog/redo03.log' to
'/nfsanf/oracopy/redo03.log';

Database altered.

SQL> alter database open resetlogs;

Database altered.

SQL> show pdbs

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>CON_NAME</th>
<th>OPEN MODE</th>
<th>RESTRICTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>READ ONLY</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>NTAP1_PDB1</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>4</td>
<td>NTAP1_PDB2</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>5</td>
<td>NTAP1_PDB3</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
</tbody>
</table>

19. Validate the database structure restored to new host as well as the test row we have inserted before primary VLDB failure.

SQL> select name from v$datafile;
SQL> alter session set container=ntap1_pdb1;
Session altered.

SQL> select * from test;

ID
------

DT
----------------------
------

EVENT
----------------------
------

1
21-MAR-24 03:15:03.000000 PM
Test oracle incremental merge switch to copy

2
22-MAR-24 02:22:06.000000 PM
Test recovery on a new Azure VM host with image copy on ANF

20. Drop invalid tempfiles and add new tempfiles to temp tablespaces.

SQL> select name from v$tempfile;
/u02/oradata/NTAP1/NTAP1_pdb1/temp01.dbf
/u02/oradata/NTAP1/NTAP1_pdb1/temp02.dbf

SQL> alter tablespace temp add tempfile
'/nfsanf/oracopy/ntap1_pdb1_temp01.dbf' size 100M;
Tablespace altered.

SQL> select name from v$tempfile;
NAME
---------------------------------
/u02/oradata/NTAP1/NTAP1_pdb1/temp01.dbf
/u02/oradata/NTAP1/NTAP1_pdb1/temp02.dbf
/nfsanf/oracopy/ntap1_pdb1_temp01.dbf

SQL> alter database tempfile
'/u02/oradata/NTAP1/NTAP1_pdb1/temp01.dbf' offline;
Database altered.

SQL> alter database tempfile
'/u02/oradata/NTAP1/NTAP1_pdb1/temp01.dbf' drop;
Database altered.

SQL> alter database tempfile
'/u02/oradata/NTAP1/NTAP1_pdb1/temp02.dbf' offline;
Database altered.

SQL> alter database tempfile
'/u02/oradata/NTAP1/NTAP1_pdb1/temp02.dbf' drop;
Database altered.

SQL> select name from v$tempfile;
NAME
---------------------------------
/nfsanf/oracopy/ntap1_pdb1_temp01.dbf

SQL>
21. Other post recovery tasks

- Add ANF NFS mount to fstab so that the NFS file system will be mounted when DB server host rebooted.

  As azureuser, sudo vi /etc/fstab and add following entry:

  172.30.136.68:/ora-01-u02-copy /nfsanf nfs
timeo=600,rsize=262144,wsize=262144,nodir 0 0

- Update the Oracle init file from primary database init file backup that is restored to /tmp/archive and create spfile as needed.

This completes the Oracle VLDB database recovery from backup image copy on ANF NFS file system to a standby DB server host.

Where to find additional information

To learn more about the information described in this document, review the following documents and/or websites:

- RMAN: Merged Incremental Backup Strategies (Doc ID 745798.1)
  https://support.oracle.com/knowledge/Oracle%20Database%20Products/745798_1.html

- RMAN Backup and Recovery User’s Guide

- Azure NetApp Files

TR-4987: Simplified, Automated Oracle Deployment on Azure NetApp Files with NFS

Allen Cao, Niyaz Mohamed, NetApp

Purpose

Running performance-intensive and latency-sensitive Oracle workloads in the cloud can be challenging. Azure NetApp Files (ANF) makes it easy for enterprise line-of-business (LOB) and storage professionals to migrate and run demanding Oracle workloads without code change. Azure NetApp Files is widely used as the underlying shared file-storage service in various scenarios, such as new deployment or migration (lift and shift) of Oracle databases from on-premises to Azure.

This documentation demonstrates the simplified deployment of Oracle databases in Azure NetApp files via NFS mounts using Ansible automation. The Oracle database deploys in a container database (CDB) and
pluggable databases (PDB) configuration with Oracle dNFS protocol enabled to boost performance. Furthermore, the on-premises Oracle single instance database or PDB can be migrated into a newly deployed container database in Azure using automated PDB relocation methodology with minimal service interruption. It also provides information on fast Oracle database backup, restore, and clone with NetApp SnapCenter UI tool in Azure Cloud.

This solution addresses the following use cases:

- Automated Oracle container database deployment on Azure NetApp files
- Automated Oracle database migration between on-premises and Azure cloud

**Audience**

This solution is intended for the following people:

- A DBA who would like to deploy Oracle on Azure NetApp Files.
- A database solution architect who would like to test Oracle workloads on Azure NetApp Files.
- A storage administrator who would like to deploy and manage an Oracle database on Azure NetApp Files.
- An application owner who would like to stand up an Oracle database on Azure NetApp Files.

**Solution test and validation environment**

The testing and validation of this solution were performed in a lab setting that might not match the final deployment environment. See the section [Key Factors for Deployment Consideration] for more information.

**Architecture**
Hardware and software components

**Hardware**

<table>
<thead>
<tr>
<th>Component</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azure NetApp Files</td>
<td>Current offering in Azure by Microsoft One capacity pool with Premium service level</td>
</tr>
<tr>
<td>Azure VM for DB server</td>
<td>Standard_B4ms - 4 vCPUs, 16GiB Two Linux virtual machine instances for concurrent deployment</td>
</tr>
<tr>
<td>Azure VM for SnapCenter</td>
<td>Standard_B4ms - 4 vCPUs, 16GiB One Windows virtual machine instance</td>
</tr>
</tbody>
</table>

**Software**

<table>
<thead>
<tr>
<th>Component</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>RedHat Linux</td>
<td>RHEL Linux 8.6 (LVM) - x64 Gen2 Deployed RedHat subscription for testing</td>
</tr>
<tr>
<td>Windows Server</td>
<td>2022 DataCenter; Azure Edition Hotpatch - x64 Gen2 Hosting SnapCenter server</td>
</tr>
<tr>
<td>Oracle Database</td>
<td>Version 19.18 Applied RU patch p34765931_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle OPatch</td>
<td>Version 12.2.0.1.36 Latest patch p6880880_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>SnapCenter Server</td>
<td>Version 5.0 Workgroup deployment</td>
</tr>
<tr>
<td>Open JDK</td>
<td>Version java-11-openjdk SnapCenter plugin requirement on DB VMs</td>
</tr>
<tr>
<td>NFS</td>
<td>Version 3.0 Oracle dNFS enabled</td>
</tr>
<tr>
<td>Ansible</td>
<td>core 2.16.2 Python 3.6.8</td>
</tr>
</tbody>
</table>

**Oracle database configuration in the lab environment**

<table>
<thead>
<tr>
<th>Server</th>
<th>Database</th>
<th>DB Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora-01</td>
<td>NTAP1(NTAP1_PDB1,NTAP1_PDB2,NTAP1_PDB3)</td>
<td>/u01, /u02, /u03 NFS mounts on ANF capacity pool</td>
</tr>
<tr>
<td>ora-02</td>
<td>NTAP2(NTAP2_PDB1,NTAP2_PDB2,NTAP2_PDB3)</td>
<td>/u01, /u02, /u03 NFS mounts on ANF capacity pool</td>
</tr>
</tbody>
</table>

**Key factors for deployment consideration**

- **Oracle database storage layout.** In this automated Oracle deployment, we provision three database volumes for each database to host Oracle binary, data, and logs by default. The volumes are mounted on Oracle DB server as /u01 - binary, /u02 - data, /u03 - logs via NFS. Dual control files are configured on /u02 and /u03 mount points for redundancy.

- **Multiple DB servers deployment.** The automation solution can deploy an Oracle container database to multiple DB servers in a single Ansible playbook run. Regardless of the number of DB servers, the playbook execution remains the same. You can deploy multiple container databases to a single VM.
instance by repeating the deployment with different database instance IDs (Oracle SID). But ensure there is sufficient memory on the host to support deployed databases.

- **dNFS configuration.** By using dNFS (available since Oracle 11g), an Oracle database running on an Azure Virtual Machine can drive significantly more I/O than the native NFS client. Automated Oracle deployment configures dNFS on NFSv3 by default.

- **Allocate large size volume to speed up deployment.** ANF file system IO throughput is regulated based on the size of volume. For initial deployment, allocate large size volumes can speed up the deployment. The volumes subsequently can be downsized dynamically without application impact.

- **Database backup.** NetApp provides a SnapCenter software suite for database backup, restore, and cloning with a user-friendly UI interface. NetApp recommends implementing such a management tool to achieve fast (under a minute) snapshot backup, quick (minutes) database restore, and database clone.

### Solution deployment

The following sections provide step-by-step procedures for automated Oracle 19c deployment and database migration on Azure NetApp Files with directly mounted database volumes via NFS to Azure VMs.

### Prerequisites for deployment
Deployment requires the following prerequisites.

1. An Azure account has been set up, and the necessary VNet and network segments have been created within your Azure account.

2. From the Azure cloud portal, deploy Azure Linux VMs as Oracle DB servers. Create an Azure NetApp Files capacity pool and database volumes for Oracle database. Enable VM SSH private/public key authentication for azureuser to DB servers. See the architecture diagram in the previous section for details about the environment setup. Also referred to Step-by-Step Oracle deployment procedures on Azure VM and Azure NetApp Files for detailed information.

   For Azure VMs deployed with local disk redundancy, ensure that you have allocated at least 128G in the VM root disk to have sufficient space to stage Oracle installation files and add OS swap file. Expand /tmplv and /rootlv OS partition accordingly. Ensure the database volume naming follows the VMname-u01, VMname-u02, and VMname-u03 convention.

   ```
   sudo lvresize -r -L +20G /dev/mapper/rootvg-rootlv
   ```

   ```
   sudo lvresize -r -L +10G /dev/mapper/rootvg-tmplv
   ```

3. From the Azure cloud portal, provision a Windows server to run the NetApp SnapCenter UI tool with the latest version. Refer to the following link for details: Install the SnapCenter Server

4. Provision a Linux VM as the Ansible controller node with the latest version of Ansible and Git installed. Refer to the following link for details: Getting Started with NetApp solution automation in section - Setup the Ansible Control Node for CLI deployments on RHEL / CentOS or Setup the Ansible Control Node for CLI deployments on Ubuntu / Debian.

   The Ansible controller node can locate either on-premises or in Azure cloud as far as it can reach Azure DB VMs via ssh port.

5. Clone a copy of the NetApp Oracle deployment automation toolkit for NFS.

   ```
   git clone https://bitbucket.ngage.netapp.com/scm/ns-bb/na_oracle_deploy_nfs.git
   ```

6. Stage following Oracle 19c installation files on Azure DB VM /tmp/archive directory with 777 permission.

   ```
   installer_archives:
   - "LINUX.X64_193000_db_home.zip"
   - "p34765931_190000_Linux-x86-64.zip"
   - "p6880880_190000_Linux-x86-64.zip"
   ```
7. Watch the following video:

Simplified and automated Oracle deployment on Azure NetApp Files with NFS

Automation parameter files

Ansible playbook executes database installation and configuration tasks with predefined parameters. For this Oracle automation solution, there are three user-defined parameter files that need user input before playbook execution.

- hosts - define targets that the automation playbook is running against.
- vars/vars.yml - the global variable file that defines variables that apply to all targets.
- host_vars/host_name.yml - the local variable file that defines variables that apply only to a named target. In our use case, these are the Oracle DB servers.

In addition to these user-defined variable files, there are several default variable files that contain default parameters that do not require change unless necessary. The following sections show how to configure the user-defined variable files.

Parameter files configuration
1. Ansible target hosts file configuration:

```
# Enter Oracle servers names to be deployed one by one, follow by each Oracle server public IP address, and ssh private key of azureuser for the server.
[oracle]
ora-01 ansible_host=10.61.180.21 ansible_ssh_private_key_file=ora-01.pem
ora-02 ansible_host=10.61.180.23 ansible_ssh_private_key_file=ora-02.pem
```

2. Global vars/vars.yml file configuration
### Oracle 19c deployment user configuration variables

### Consolidate all variables from ANF, linux and oracle

### ANF env specific config variables

Prerequisite to create three volumes in NetApp storage pool from cloud dashboard with following naming convention:
- `db_hostname-u01` - Oracle binary
- `db_hostname-u02` - Oracle data
- `db_hostname-u03` - Oracle redo

It is important to strictly follow the name convention or the automation will fail.

NFS lif ip address to access database volumes in ANF storage pool (retrievable from cloud dashboard):
- `nfs_lif: 172.30.136.68`

### Linux env specific config variables

- `redhat_sub_username: XXXXXXXX`
- `redhat_sub_password: XXXXXXXX`

### DB env specific install and config variables

- `db_domain: solutions.netapp.com`
- `initial_pwd_all: XXXXXXXX`

3. Local DB server `host_vars/host_name.yml` configuration such as `ora_01.yml`, `ora_02.yml`...
# User configurable Oracle host specific parameters

# Enter container database SID. By default, a container DB is created with 3 PDBs within the CDB
oracle_sid: NTAP1

# Enter database shared memory size or SGA. CDB is created with SGA at 75% of memory_limit, MB. The grand total of SGA should not exceed 75% available RAM on node.
memory_limit: 8192

Playbook execution
There are a total of five playbooks in the automation toolkit. Each performs different task blocks and serves different purposes.

0-all_playbook.yml - execute playbooks from 1-4 in one playbook run.
1-ansible_requirements.yml - set up Ansible controller with required libs and collections.
2-linux_config.yml - execute Linux kernel configuration on Oracle DB servers.
4-oracle_config.yml - install and configure Oracle on DB servers and create a container database.
5-destroy.yml - optional to undo the environment to dismantle all.

There are three options to run the playbooks with the following commands.

1. Execute all deployment playbooks in one combined run.

   ```bash
   ansible-playbook -i hosts 0-all_playbook.yml -u azureuser -e @vars/vars.yml
   ```

2. Execute playbooks one at a time with the number sequence from 1-4.

   ```bash
   ansible-playbook -i hosts 1-ansible_requirements.yml -u azureuser -e @vars/vars.yml
   ```

   ```bash
   ansible-playbook -i hosts 2-linux_config.yml -u azureuser -e @vars/vars.yml
   ```

   ```bash
   ansible-playbook -i hosts 4-oracle_config.yml -u azureuser -e @vars/vars.yml
   ```

3. Execute 0-all_playbook.yml with a tag.

   ```bash
   ansible-playbook -i hosts 0-all_playbook.yml -u azureuser -e @vars/vars.yml -t ansible_requirements
   ```

   ```bash
   ansible-playbook -i hosts 0-all_playbook.yml -u azureuser -e @vars/vars.yml -t linux_config
   ```
4. Undo the environment

```
ansible-playbook -i hosts 5-destroy.yml -u azureuser -e
@vars/vars.yml
```

Post execution validation
After the playbook run, login to the Oracle DB server VM to validate that Oracle is installed and configured and a container database is created successfully. Following is an example of Oracle database validation on host ora-01.

1. Validate NFS mounts

```bash
[azureuser@ora-01 ~]$ cat /etc/fstab
#
#/etc/fstab
#/ Created by anaconda on Thu Sep 14 11:04:01 2023
#/ Accessible filesystems, by reference, are maintained under
#/ '/dev/disk/'.
#/ See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for
#/ more info.
#/ After editing this file, run 'systemctl daemon-reload' to update
# systemd
#/ units generated from this file.
#/ /dev/mapper/rootvg-rootlv / xfs defaults 0 0
# UUID=268633bd-f9bb-446d-9a1d-8fca4609a1e1 /boot
# xfs defaults 0 0
# UUID=89D8-B037 /boot/efi vfat
# defaults,uid=0,gid=0,umask=077,shortname=winnt 0 2
#/dev/mapper/rootvg-homelv /home xfs defaults 0 0
# /dev/mapper/rootvg-tmplv /tmp xfs defaults 0 0
# /dev/mapper/rootvg-usrlv /usr xfs defaults 0 0
# /dev/mapper/rootvg-varlv /var xfs defaults 0 0
# /mnt/swapfile swap swap defaults 0 0
# 172.30.136.68:/ora-01-u01 /u01 nfs
# rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536 0 0
# 172.30.136.68:/ora-01-u02 /u02 nfs
# rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536 0 0
# 172.30.136.68:/ora-01-u03 /u03 nfs
# rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536 0 0

[azureuser@ora-01 ~]$ df -h
Filesystem Size Used Avail Use% Mounted on
devtmpfs 7.7G 0 7.7G 0% /dev
```
tmpfs  7.8G  0  7.8G  0% /dev/shm
tmpfs  7.8G  8.6M  7.7G  1% /run
tmpfs  7.8G  0  7.8G  0% /sys/fs/cgroup
/dev/mapper/rootvg-rootlv  22G  17G  5.8G  74% /
/dev/mapper/rootvg-usrlv  10G  2.0G  8.1G  20% /usr
/dev/mapper/rootvg-varlv  8.0G  890M  7.2G  11% /var
/dev/sda1  496M  106M  390M  22% /boot
/dev/mapper/rootvg-homelv  1014M  40M  975M  4% /home
/dev/sda15  495M  5.9M  489M  2% /boot/efi
/dev/mapper/rootvg-tmplv  12G  8.4G  3.7G  70% /tmp
tmpfs  1.6G  0  1.6G  0% /run/user/54321
172.30.136.68:/ora-01-u01  500G  11G  490G  3% /u01
172.30.136.68:/ora-01-u03  250G  1.2G  249G  1% /u03
172.30.136.68:/ora-01-u02  250G  7.1G  243G  3% /u02
tmpfs  1.6G  0  1.6G  0% /run/user/1000

2. Validate Oracle listener

[azureuser@ora-01 ~]$ sudo su
[root@ora-01 azureuser]# su - oracle
Last login: Thu Feb 1 16:13:44 UTC 2024
[oracle@ora-01 ~]$ lsnrctl status listener.ntap1

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 01-FEB-2024 16:25:37

Copyright (c) 1991, 2022, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=ora-
01.internal.cloudapp.net)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias LISTENER.NTAP1
Version TNSLSNR for Linux: Version 19.0.0.0.0 -
Production
Start Date 01-FEB-2024 16:13:49
Uptime 0 days 0 hr. 11 min. 49 sec
Trace Level off
Security ON: Local OS Authentication
SNMP OFF
Listener Parameter File /u01/app/oracle/product/19.0.0/NTAP1/network/admin/listener.ora
Listener Log File /u01/app/oracle/diag/tnslsnr/ora-
01/listener.ntap1/alert/log.xml
Listening Endpoints Summary...
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=ora-01.hr2z2nbmhnqutsxgscjtuxizd.jx.internal.cloudapp.net)(PORT=1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=ora-01.hr2z2nbmhnqutsxgscjtuxizd.jx.internal.cloudapp.net)(PORT=5500))(Security=(my_wallet_directory=/u01/app/oracle/product/19.0.0/NTAP1/admin/NTAP1/adb_wallet))(Presentation=HTTP)(Session=RAW))

Services Summary...
Service "104409ac02da6352e063bb891eacf34a.solutions.netapp.com" has 1 instance(s).
  Instance "NTAP1", status READY, has 1 handler(s) for this service...
Service "104412c14c2c63cbe063bb891eacf64d.solutions.netapp.com" has 1 instance(s).
  Instance "NTAP1", status READY, has 1 handler(s) for this service...
Service "1044174670ad63ffe063bb891eacf6b34.solutions.netapp.com" has 1 instance(s).
  Instance "NTAP1", status READY, has 1 handler(s) for this service...
Service "NTAP1.solutions.netapp.com" has 1 instance(s).
  Instance "NTAP1", status READY, has 1 handler(s) for this service...
Service "NTAP1XDB.solutions.netapp.com" has 1 instance(s).
  Instance "NTAP1", status READY, has 1 handler(s) for this service...
Service "ntap1_pdb1.solutions.netapp.com" has 1 instance(s).
  Instance "NTAP1", status READY, has 1 handler(s) for this service...
Service "ntap1_pdb2.solutions.netapp.com" has 1 instance(s).
  Instance "NTAP1", status READY, has 1 handler(s) for this service...
Service "ntap1_pdb3.solutions.netapp.com" has 1 instance(s).
  Instance "NTAP1", status READY, has 1 handler(s) for this service...

The command completed successfully

3. Validate Oracle database and dNFS

[oracle@ora-01 ~]$ cat /etc/oratab
#
# This file is used by ORACLE utilities. It is created by root.sh
# and updated by either Database Configuration Assistant while creating
# a database or ASM Configuration Assistant while creating ASM
instance.

# A colon, ':', is used as the field terminator. A new line
# terminates the entry. Lines beginning with a pound sign, '#', are comments.
# Entries are of the form:
#   $ORACLE_SID:$ORACLE_HOME:<N|Y>:  
# The first and second fields are the system identifier and home
# directory of the database respectively. The third field indicates
# to the dbstart utility that the database should, "Y", or should not,
# "N", be brought up at system boot time.
# Multiple entries with the same $ORACLE_SID are not allowed.
# NTAP1:/u01/app/oracle/product/19.0.0/NTAP1:Y

[oracle@ora-01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Feb 1 16:37:51 2024
Version 19.18.0.0.0
Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0

SQL> select name, open_mode, log_mode from v$database;

NAME      OPEN_MODE            LOG_MODE
--------- -------------------- ------------
NTAP1     READ WRITE           ARCHIVELOG

SQL> show pdbs

CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
2 PDB$SEED                       READ ONLY  NO
3 NTAP1_PDB1                     READ WRITE NO
4 NTAP1_PDB2                     READ WRITE NO
SQL> select name from v$datafile;

NAME
-----------------------------------------------
/u02/oradata/NTAP1/system01.dbf
/u02/oradata/NTAP1/syseq01.dbf
/u02/oradata/NTAP1/undotbs01.dbf
/u02/oradata/NTAP1/pdbseed/system01.dbf
/u02/oradata/NTAP1/pdbseed/syseq01.dbf
/u02/oradata/NTAP1/users01.dbf
/u02/oradata/NTAP1/pdbseed/undotbs01.dbf
/u02/oradata/NTAP1/NTAP1_pdb1/system01.dbf
/u02/oradata/NTAP1/NTAP1_pdb1/syseq01.dbf
/u02/oradata/NTAP1/NTAP1_pdb1/undotbs01.dbf
/u02/oradata/NTAP1/NTAP1_pdb1/users01.dbf

NAME
-----------------------------------------------
/u02/oradata/NTAP1/NTAP1_pdb2/system01.dbf
/u02/oradata/NTAP1/NTAP1_pdb2/syseq01.dbf
/u02/oradata/NTAP1/NTAP1_pdb2/undotbs01.dbf
/u02/oradata/NTAP1/NTAP1_pdb2/users01.dbf
/u02/oradata/NTAP1/NTAP1_pdb3/system01.dbf
/u02/oradata/NTAP1/NTAP1_pdb3/syseq01.dbf
/u02/oradata/NTAP1/NTAP1_pdb3/undotbs01.dbf
/u02/oradata/NTAP1/NTAP1_pdb3/users01.dbf

19 rows selected.

SQL> select name from v$controlfile;

NAME
-----------------------------------------------
/u02/oradata/NTAP1/control01.ctl
/u03/orareco/NTAP1/control02.ctl

SQL> select member from v$logfile;

MEMBER
-----------------------------------------------
/u03/orareco/NTAP1/onlinelg/redo03.log
4. Login to Oracle Enterprise Manager Express to validate database.
Migrate Oracle database to Azure

Oracle database migration from on-premises to the cloud is a heavy-lifting. Using the right strategy and automation can smooth the process and minimize service interruption and downtime. Follow this detailed instruction Database migration from on-premises to Azure cloud to guide your database migration journey.

Oracle backup, restore, and clone with SnapCenter

NetApp recommends SnapCenter UI tool to manage Oracle database deployed in Azure cloud. Please refer to TR-4988: Oracle Database Backup, Recovery, and Clone on ANF with SnapCenter for details.
Where to find additional information

To learn more about the information described in this document, review the following documents and/or websites:

• Oracle Database Backup, Recovery, and Clone on ANF with SnapCenter

  Oracle Database Backup, Recovery, and Clone on ANF with SnapCenter

• Azure NetApp Files


• Deploying Oracle Direct NFS


• Installing and Configuring Oracle Database Using Response Files


TR-4986: Simplified, Automated Oracle Deployment on Amazon FSx ONTAP with iSCSI

Allen Cao, Niyaz Mohamed, NetApp

Purpose

Amazon FSx for NetApp ONTAP is a storage service that allows you to launch and run fully managed NetApp ONTAP file systems in the AWS Cloud. It provides the familiar features, performance, capabilities, and APIs of NetApp file systems with the agility, scalability, and simplicity of a fully managed AWS service. It empowers you to run the most demanding database workload, such as Oracle, in the AWS cloud with peace of mind.

This documentation demonstrates the simplified deployment of Oracle databases in an Amazon FSx ONTAP file system using Ansible automation. The Oracle database is deployed in a standalone ReStart configuration with iSCSI protocol for data access and Oracle ASM for database storage disks management. It also provides information on Oracle database backup, restore, and clone using the NetApp SnapCenter UI tool for storage-efficient database operation in AWS Cloud.

This solution addresses the following use cases:

• Automated Oracle database deployment on Amazon FSx ONTAP file system
• Oracle database backup and restore on Amazon FSx ONTAP file system using NetApp SnapCenter tool
• Oracle database clone for dev/test or other use cases on Amazon FSx ONTAP file system using NetApp SnapCenter tool

Audience

This solution is intended for the following people:

• A DBA who would like to deploy Oracle on Amazon FSx ONTAP file system.
• A database solution architect who would like to test Oracle workloads on Amazon FSx ONTAP file system.
• A storage administrator who would like to deploy and manage an Oracle database on Amazon FSx ONTAP file system.
• An application owner who would like to stand up an Oracle database on Amazon FSx ONTAP file system.

**Solution test and validation environment**

The testing and validation of this solution were performed in a lab setting that might not match the final deployment environment. See the section [Key Factors for Deployment Consideration] for more information.

**Architecture**

**Simplified, automated Oracle deployment on Amazon FSx ONTAP with iSCSI**

**Hardware and software components**

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon FSx ONTAP storage</td>
<td>RedHat Linux</td>
</tr>
<tr>
<td>Current version offered by AWS</td>
<td>RHEL-8.6, 4.18.0-372.9.1.el8.x86_64 kernel</td>
</tr>
<tr>
<td>One FSx HA cluster in the same VPC and availability zone</td>
<td>Windows Server</td>
</tr>
<tr>
<td>EC2 instance for compute</td>
<td></td>
</tr>
</tbody>
</table>
Oracle Grid Infrastructure  Version 19.18  Applied RU patch p34762026_190000_Linux-x86-64.zip

Oracle Database  Version 19.18  Applied RU patch p34765931_190000_Linux-x86-64.zip

Oracle OPatch  Version 12.2.0.1.36  Latest patch p6880880_190000_Linux-x86-64.zip

SnapCenter Server  Version 4.9P1  Workgroup deployment

Open JDK  Version java-1.8.0-openjdk.x86_64  SnapCenter plugin requirement on DB VMs

Oracle database configuration in the lab environment

<table>
<thead>
<tr>
<th>Server</th>
<th>Database</th>
<th>DB Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora_01</td>
<td>NTAP1(NTAP1_PDB1,NTAP1_PDB2,NTAP1_PDB3)</td>
<td>iSCSI luns on Amazon FSx ONTAP file system</td>
</tr>
<tr>
<td>ora_02</td>
<td>NTAP2(NTAP2_PDB1,NTAP2_PDB2,NTAP2_PDB3)</td>
<td>iSCSI luns on Amazon FSx ONTAP file system</td>
</tr>
</tbody>
</table>

Key factors for deployment consideration

- **Oracle database storage layout.** In this automated Oracle deployment, we provision four database volumes to host Oracle binary, data, and logs by default. A single lun in a volume allocates to Oracle binary. We then create two ASM disk groups from data and logs luns. Within the +DATA asm disk group, we provision two data volumes with two luns in a volume. Within the +LOGS asm disk group, we create two luns in a log volume. Multiple luns laid out within an ONTAP volume provides better performance in general.

- **Multiple DB servers deployment.** The automation solution can deploy an Oracle container database to multiple DB servers in a single Ansible playbook run. Regardless of the number of DB servers, the playbook execution remains the same. You can deploy multiple container databases to a single EC2 instance with different database instance IDs (Oracle SID). But ensure there is sufficient memory on the host to support deployed databases.

- **iSCSI configuration.** The EC2 instance database server connects to FSx storage with the iSCSI protocol. EC2 instances generally deploy with a single network interface or ENI. The single NIC interface carries both iSCSI and application traffic. It is important to gauge the Oracle database peak I/O throughput requirement by carefully analyzing the Oracle AWR report in order to choose the right EC2 compute instance that meets both application and iSCSI traffic-throughput requirements. Also, AWS EC2 generally limits each TCP flow to 5 Gbps. Each iSCSI path provides 5 Gbps (625 MBps) of bandwidth, and multiple iSCSI connections may be required to support higher throughput requirements.

- **Oracle ASM redundancy level to use for each Oracle ASM disk group that you create.** Because the Amazon FSx ONTAP is HA enabled for data protection at the cluster disk level, you should use External Redundancy, which means that the option does not allow Oracle ASM to mirror the contents of the disk group.

- **Database backup.** NetApp provides a SnapCenter software suite for database backup, restore, and cloning with a user-friendly UI interface. NetApp recommends implementing such a management tool to
achieve fast (under a minute) SnapShot backup, quick (minutes) database restore, and database clone.

**Solution deployment**

The following sections provide step-by-step procedures for automated Oracle 19c deployment and protection on Amazon FSx ONTAP file system with directly mounted database luns via iSCSI to EC2 instance VM in a single node Restart configuration with Oracle ASM as database volume manager.

**Prerequisites for deployment**
Deployment requires the following prerequisites.

1. An AWS account has been set up, and the necessary VPC and network segments have been created within your AWS account.

2. From the AWS EC2 console, deploy EC2 Linux instances as Oracle DB servers. Enable SSH private/public key authentication for ec2-user. See the architecture diagram in the previous section for details about the environment setup. Also review the User Guide for Linux instances for more information.

3. From the AWS FSx console, provision an Amazon FSx ONTAP file system that meets the requirements. Review the documentation Creating FSx for ONTAP file systems for step-by-step instructions.

4. Steps 2 and 3 can be performed using the following Terraform automation toolkit, which creates an EC2 instance named ora_01 and an FSx file system named fsx_01. Review the instruction carefully and change the variables to suit your environment before execution. The template can be easily revised for your own deployment requirements.

```bash
git clone https://github.com/NetApp-Automation/na_aws_fsx_ec2_deploy.git
```

5. Provision an EC2 Linux instance as the Ansible controller node with the latest version of Ansible and Git installed. Refer to the following link for details: Getting Started with NetApp solution automation in section - Setup the Ansible Control Node for CLI deployments on RHEL / CentOS or Setup the Ansible Control Node for CLI deployments on Ubuntu / Debian.

6. Provision a Windows server to run the NetApp SnapCenter UI tool with the latest version. Refer to the following link for details: Install the SnapCenter Server

7. Clone a copy of the NetApp Oracle deployment automation toolkit for iSCSI.

```bash
git clone https://bitbucket.ngage.netapp.com/scm/ns-bb/na_oracle_deploy_iscsi.git
```

8. Stage following Oracle 19c installation files on EC2 instances /tmp/archive directory.

```bash
installer_archives:
  - "LINUX.X64_193000_grid_home.zip"
  - "p34762026_190000_Linux-x86-64.zip"
  - "LINUX.X64_193000_db_home.zip"
  - "p34765931_190000_Linux-x86-64.zip"
  - "p6880880_190000_Linux-x86-64.zip"
```

Ensure that you have allocated at least 50G in Oracle VM root volume to have sufficient space to stage Oracle installation files.

9. Watch the following video:
Simplified and automated Oracle deployment on Amazon FSx ONTAP with iSCSI

Automation parameter files

Ansible playbook executes database installation and configuration tasks with predefined parameters. For this Oracle automation solution, there are three user-defined parameter files that need user input before playbook execution.

- **hosts** - define targets that the automation playbook is running against.
- **vars/vars.yml** - the global variable file that defines variables that apply to all targets.
- **host_vars/host_name.yml** - the local variable file that defines variables that apply only to a named target. In our use case, these are the Oracle DB servers.

In addition to these user-defined variable files, there are several default variable files that contain default parameters that do not require change unless necessary. The following sections show how to configure the user-defined variable files.

Parameter files configuration
1. Ansible target hosts file configuration:

```yaml
# Enter Amazon FSx ONTAP management IP address
[ontap]
172.16.9.32

# Enter name for ec2 instance (not default IP address naming) to be deployed one by one, follow by ec2 instance IP address, and ssh private key of ec2-user for the instance.
[oracle]
ora_01 ansible_host=10.61.180.21 ansible_ssh_private_key_file =ora_01.pem
ora_02 ansible_host=10.61.180.23 ansible_ssh_private_key_file =ora_02.pem
```

2. Global vars/vars.yml file configuration

```yaml
# Enter the supported ONTAP platform: on-prem, aws-fsx.
ontap_platform: aws-fsx

# Enter ONTAP cluster management user credentials
username: "fsxadmin"
password: "xxxxxxxxx"
```
# Enter RHEL subscription to enable repo
redhat_sub_username: xxxxxxxx
redhat_sub_password: "xxxxxxxx"

# Enter Database domain name
db_domain: solutions.netapp.com

# Enter initial password for all required Oracle passwords. Change them after installation.
initial_pwd_all: xxxxxxxx

3. Local DB server host_vars/host_name.yml configuration such as ora_01.yml, ora_02.yml ...

# User configurable Oracle host specific parameters

# Enter container database SID. By default, a container DB is created with 3 PDBs within the CDB
oracle_sid: NTAP1

# Enter database shared memory size or SGA. CDB is created with SGA at 75% of memory_limit, MB. The grand total of SGA should not exceed 75% available RAM on node.
memory_limit: 8192

Playbook execution
There are a total of six playbooks in the automation toolkit. Each performs different task blocks and serves different purposes.

- **0-all_playbook.yml** - execute playbooks from 1-4 in one playbook run.
- **1-ansible_requirements.yml** - set up Ansible controller with required libs and collections.
- **2-linux_config.yml** - execute Linux kernel configuration on Oracle DB servers.
- **3-ontap_config.yml** - configure ONTAP svm/volumes/luns for Oracle database and grant DB server access to luns.
- **4-oracle_config.yml** - install and configure Oracle on DB servers for grid infrastructure and create a container database.
- **5-destroy.yml** - optional to undo the environment to dismantle all.

There are three options to run the playbooks with the following commands.

1. **Execute all deployment playbooks in one combined run.**

   ```
   ansible-playbook -i hosts 0-all_playbook.yml -u ec2-user -e @vars/vars.yml
   ```

2. **Execute playbooks one at a time with the number sequence from 1-4.**

   ```
   ansible-playbook -i hosts 1-ansible_requirements.yml -u ec2-user -e @vars/vars.yml
   ```

   ```
   ansible-playbook -i hosts 2-linux_config.yml -u ec2-user -e @vars/vars.yml
   ```

   ```
   ansible-playbook -i hosts 3-ontap_config.yml -u ec2-user -e @vars/vars.yml
   ```

   ```
   ansible-playbook -i hosts 4-oracle_config.yml -u ec2-user -e @vars/vars.yml
   ```

3. **Execute 0-all_playbook.yml with a tag.**

   ```
   ansible-playbook -i hosts 0-all_playbook.yml -u ec2-user -e @vars/vars.yml
   ```
```bash
ansible-playbook -i hosts 0-all_playbook.yml -u ec2-user -e @vars/vars.yml -t ansible_requirements

ansible-playbook -i hosts 0-all_playbook.yml -u ec2-user -e @vars/vars.yml -t linux_config

ansible-playbook -i hosts 0-all_playbook.yml -u ec2-user -e @vars/vars.yml -t ontap_config

ansible-playbook -i hosts 0-all_playbook.yml -u ec2-user -e @vars/vars.yml -t oracle_config

4. Undo the environment

ansible-playbook -i hosts 5-destroy.yml -u ec2-user -e @vars/vars.yml

Post execution validation
After the playbook run, login to the Oracle DB server as oracle user to validate that Oracle grid infrastructure and database are created successfully. Following is an example of Oracle database validation on host ora_01.

1. Validate Oracle container database on EC2 instance

```
[admin@ansiblectl na_oracle_deploy_iscsi]$ ssh -i ora_01.pem ec2-user@172.30.15.40
Last login: Fri Dec 8 17:14:21 2023 from 10.61.180.18
[ec2-user@ip-172-30-15-40 ~]$ uname -a
Linux ip-172-30-15-40.ec2.internal 4.18.0-372.9.1.el8.x86_64 #1 SMP
Fri Apr 15 22:12:19 EDT 2022 x86_64 x86_64 x86_64 GNU/Linux

[ec2-user@ip-172-30-15-40 ~]$ sudo su
[root@ip-172-30-15-40 ec2-user]# su - oracle
Last login: Fri Dec 8 16:25:52 UTC 2023 on pts/0
[oracle@ip-172-30-15-40 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Dec 8 18:18:20 2023
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.18.0.0.0

SQL> select name, open_mode, log_mode from v$database;
NAME      OPEN_MODE            LOG_MODE
--------- -------------------- ------------
NTAP1     READ WRITE           ARCHIVELOG

SQL> show pdbs

CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
  2 PDB$SEED                       READ ONLY  NO
  3 NTAP1_PDB1                     READ WRITE NO
  4 NTAP1_PDB2                     READ WRITE NO
  5 NTAP1_PDB3                     READ WRITE NO

SQL> select name from v$datafile;
NAME
778
```
+DATA/NTAP1/DATAFILE/system.257.1155055419
+DATA/NTAP1/DATAFILE/sysaux.258.1155055463
+DATA/NTAP1/DATAFILE/undotbs1.259.1155055489
+DATA/NTAP1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/system.266.1155056241
+DATA/NTAP1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/sysaux.267.1155056241
+DATA/NTAP1/DATAFILE/users.260.11550555489
+DATA/NTAP1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/undotbs1.268.1155056241
+DATA/NTAP1/0C03AAFA7C6FD2E5E063280F1EACFBE0/DATAFILE/system.272.1155057059
+DATA/NTAP1/0C03AAFA7C6FD2E5E063280F1EACFBE0/DATAFILE/sysaux.273.1155057059
+DATA/NTAP1/0C03AAFA7C6FD2E5E063280F1EACFBE0/DATAFILE/undotbs1.271.1155057059
+DATA/NTAP1/0C03AAFA7C6FD2E5E063280F1EACFBE0/DATAFILE/users.275.1155057075

NAME

+DATA/NTAP1/0C03AC0089ACD352E063280F1EAC12BD/DATAFILE/system.277.1155057075
+DATA/NTAP1/0C03AC0089ACD352E063280F1EAC12BD/DATAFILE/sysaux.278.1155057075
+DATA/NTAP1/0C03AC0089ACD352E063280F1EAC12BD/DATAFILE/undotbs1.276.1155057075
+DATA/NTAP1/0C03AC0089ACD352E063280F1EAC12BD/DATAFILE/users.280.1155057091
+DATA/NTAP1/0C03ACEABA54D38EE6063280F1EACE573/DATAFILE/system.282.1155057091
+DATA/NTAP1/0C03ACEABA54D38EE6063280F1EACE573/DATAFILE/sysaux.283.1155057091
+DATA/NTAP1/0C03ACEABA54D38EE6063280F1EACE573/DATAFILE/undotbs1.281.1155057091
+DATA/NTAP1/0C03ACEABA54D38EE6063280F1EACE573/DATAFILE/users.285.1155057105

19 rows selected.

SQL> select name from v$controlfile;

NAME
2. Validate Oracle listener.

```bash
[oracle@ip-172-30-15-40 ~]$ lsnrctl status listener

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 08-DEC-2023 18:20:24
Copyright (c) 1991, 2022, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=ip-172-30-15-40.ec2.internal)(PORT=1521)))
STATUS of the LISTENER
--------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 19.0.0.0.0 - Production
Start Date                08-DEC-2023 16:26:09
Uptime                    0 days 1 hr. 54 min. 14 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
```
Listener Log File

Listening Endpoints Summary...

   (DESCRIPTION=(ADDRESS= (PROTOCOL= tcp) (HOST= ip-172-30-15-40.ec2.internal) (PORT= 1521)))
   (DESCRIPTION= (ADDRESS= (PROTOCOL= ipcs) (KEY= EXTPROC1521)))
   (DESCRIPTION = (ADDRESS= (PROTOCOL= tcps) (HOST= ip-172-30-15-40.ec2.internal) (PORT= 5500)) (Security= (my_wallet_directory= /u01/app/oracle/product/19.0.0/NTAP1/admin/NTAP1/xdb_wallet)) (Presentation= HTTP) (Session= RAW))

Services Summary...

Service "+ASM" has 1 instance(s).
   Instance "+ASM", status READY, has 1 handler(s) for this service...

Service "+ASM_DATA" has 1 instance(s).
   Instance "+ASM", status READY, has 1 handler(s) for this service...

Service "+ASM_LOGS" has 1 instance(s).
   Instance "+ASM", status READY, has 1 handler(s) for this service...

Service "0c03aafa7c6fd2e5e063280f1eacfbe0.solutions.netapp.com" has 1 instance(s).
   Instance "NTAP1", status READY, has 1 handler(s) for this service...

Service "0c03ac0089acd352e063280f1eac12bd.solutions.netapp.com" has 1 instance(s).
   Instance "NTAP1", status READY, has 1 handler(s) for this service...

Service "0c03aceaba54d386e063280f1eace573.solutions.netapp.com" has 1 instance(s).
   Instance "NTAP1", status READY, has 1 handler(s) for this service...

Service "NTAP1.solutions.netapp.com" has 1 instance(s).
   Instance "NTAP1", status READY, has 1 handler(s) for this service...

Service "NTAP1XDB.solutions.netapp.com" has 1 instance(s).
   Instance "NTAP1", status READY, has 1 handler(s) for this service...

Service "ntap1_pdb1.solutions.netapp.com" has 1 instance(s).
   Instance "NTAP1", status READY, has 1 handler(s) for this service...

Service "ntap1_pdb2.solutions.netapp.com" has 1 instance(s).
   Instance "NTAP1", status READY, has 1 handler(s) for this service...

Service "ntap1_pdb3.solutions.netapp.com" has 1 instance(s).
Instance "NTAP1", status READY, has 1 handler(s) for this service...
The command completed successfully

3. Validate the grid infrastructure and resources created.

```
[oracle@ip-172-30-15-40 ~]$ asm
[oracle@ip-172-30-15-40 ~]$ crsctl check has
CRS-4638: Oracle High Availability Services is online
[oracle@ip-172-30-15-40 ~]$ crsctl stat res -t

---
<table>
<thead>
<tr>
<th>Name</th>
<th>Target</th>
<th>State</th>
<th>Server</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>details</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local Resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ora.DATA.dg</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-40</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.LISTENER.lsnr</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-40</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.LOGS.dg</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-40</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.asm</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-40</td>
<td></td>
</tr>
</tbody>
</table>

Started,STABLE
ora.ons
OFFLINE OFFLINE ip-172-30-15-40 STABLE

---
<table>
<thead>
<tr>
<th>Name</th>
<th>Target</th>
<th>State</th>
<th>Server</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ora.cssd</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-40</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.diskmon</td>
<td>OFFLINE</td>
<td>OFFLINE</td>
<td></td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.driver.afd</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-40</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.evmd</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-40</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.ntap1.db</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-40</td>
<td></td>
</tr>
</tbody>
</table>
4. Validate Oracle ASM.

```
[oracle@ip-172-30-15-40 ~]$ asmcmd
ASMCMD> lsdg
State    Type    Rebal  Sector  Logical_Sector  Block       AU
Total_MB  Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks
Voting_files  Name
MOUNTED  EXTERN  N  512    512    4096  4194304
163840    155376          0    155376       0
N  DATA/
          MOUNTED  EXTERN  N  512    512    4096  4194304
81920    80972          0    80972       0
N  LOGS/
ASMCMD> lsdsk
Path
    AFD:ORA_01_DAT1_01
    AFD:ORA_01_DAT1_03
    AFD:ORA_01_DAT2_02
    AFD:ORA_01_DAT2_04
    AFD:ORA_01_LOGS_01
    AFD:ORA_01_LOGS_02
ASMCMD> afd_state
ASMCMD-9526: The AFD state is 'LOADED' and filtering is 'ENABLED' on
host 'ip-172-30-15-40.ec2.internal'
ASMCMD> exit
```

5. Login to Oracle Enterprise Manager Express to validate database.
Oracle backup, restore, and clone with SnapCenter

Refer to TR-4979 Simplified, self-managed Oracle in VMware Cloud on AWS with guest-mounted FSx ONTAP section Oracle backup, restore, and clone with SnapCenter for details on setting up SnapCenter and executing the database backup, restore, and clone workflows.

Where to find additional information

To learn more about the information described in this document, review the following documents and/or websites:

- Amazon FSx for NetApp ONTAP
  
  https://aws.amazon.com/fsx/netapp-ontap/

- Amazon EC2
  
  https://aws.amazon.com/pm/ec2/?trk=36c6da98-7b20-48fa-8225-4784bcded9843&sc_channel=ps&s_kwcid=AL!4422i3!467723097970le!!g!aws%20ec2&ef_id=Cj0KCQiA54KfbCKARlAsJzSrdqwQrgnh6I71jiWzSeaT9Uh1+vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:G:s&s_kwcid=AL!4422i3!467723097970le!!g!aws%20ec2

- Installing Oracle Grid Infrastructure for a Standalone Server with a New Database Installation
  

- Installing and Configuring Oracle Database Using Response Files
  
  https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/installing-and-configuring-oracle-
Purpose

NetApp ASA systems deliver modern solutions to your SAN infrastructure. They simplify at scale and enable you to accelerate your business-critical applications such as databases, make sure that your data is always available (99.9999% uptime), and reduce TCO and carbon footprint. The NetApp ASA systems include A-Series models designed for the most performance-demanding applications and C-Series models optimized for cost-effective, large-capacity deployments. Together, the ASA A-Series and C-Series systems deliver exceptional performance to improve customer experience and reduce time to results, keep business-critical data available, protected, and secure, and provide more effective capacity for any workload, backed by the industry’s most effective guarantee.

This documentation demonstrates the simplified deployment of Oracle databases in a SAN environment built with ASA systems using Ansible automation. The Oracle database is deployed in a standalone ReStart configuration with iSCSI protocol for data access and Oracle ASM for database disks management on the ASA storage array. It also provides information on Oracle database backup, restore, and clone using the NetApp SnapCenter UI tool for storage-efficient database operation in NetApp ASA systems.

This solution addresses the following use cases:

• Automated Oracle database deployment in NetApp ASA systems as primary database storage
• Oracle database backup and restore in NetApp ASA systems using NetApp SnapCenter tool
• Oracle database clone for dev/test or other use cases in NetApp ASA systems using NetApp SnapCenter tool

Audience

This solution is intended for the following people:

• A DBA who would like to deploy Oracle in NetApp ASA systems.
• A database solution architect who would like to test Oracle workloads in NetApp ASA systems.
• A storage administrator who would like to deploy and manage an Oracle database on NetApp ASA systems.
• An application owner who would like to stand up an Oracle database in NetApp ASA systems.

Solution test and validation environment

The testing and validation of this solution were performed in a lab setting that might not match the final deployment environment. See the section [Key Factors for Deployment Consideration] for more information.

Architecture
**Simplified, Automated Oracle Database Deployment on NetApp ASA with iSCSI**

**Hardware and software components**

### Hardware

<table>
<thead>
<tr>
<th>Component</th>
<th>Version/Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetApp ASA A400</td>
<td>Version 9.13.1P1</td>
</tr>
<tr>
<td>UCSB-B200-M4</td>
<td>Intel® Xeon® CPU E5-2690 v4 @ 2.60GHz</td>
</tr>
<tr>
<td></td>
<td>2 NS224 shelves, 48 NVMe AFF drives with total 69.3 TiB capacity</td>
</tr>
</tbody>
</table>

### Software

<table>
<thead>
<tr>
<th>Component</th>
<th>Version/Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>RedHat Linux</td>
<td>RHEL-8.6, 4.18.0-372.9.1.el8.x86_64 kernel</td>
</tr>
<tr>
<td></td>
<td>Deployed RedHat subscription for testing</td>
</tr>
<tr>
<td>Windows Server</td>
<td>2022 Standard, 10.0.20348 Build 20348</td>
</tr>
<tr>
<td></td>
<td>Hosting SnapCenter server</td>
</tr>
<tr>
<td>Oracle Grid Infrastructure</td>
<td>Version 19.18</td>
</tr>
<tr>
<td></td>
<td>Applied RU patch p34762026_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle Database</td>
<td>Version 19.18</td>
</tr>
<tr>
<td></td>
<td>Applied RU patch p34765931_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle OPatch</td>
<td>Version 12.2.0.1.36</td>
</tr>
<tr>
<td></td>
<td>Latest patch p6880880_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>SnapCenter Server</td>
<td>Version 4.9P1</td>
</tr>
<tr>
<td></td>
<td>Workgroup deployment</td>
</tr>
<tr>
<td>VMware vSphere Hypervisor</td>
<td>version 6.5.0.20000</td>
</tr>
<tr>
<td></td>
<td>VMware Tools, Version: 11365 - Linux, 12352 - Windows</td>
</tr>
</tbody>
</table>
Oracle database configuration in the lab environment

<table>
<thead>
<tr>
<th>Server</th>
<th>Database</th>
<th>DB Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora_01</td>
<td>NTAP1(NTAP1_PDB1,NTAP1_PDB2,NTAP1_PDB3)</td>
<td>iSCSI luns on ASA A400</td>
</tr>
<tr>
<td>ora_02</td>
<td>NTAP2(NTAP2_PDB1,NTAP2_PDB2,NTAP2_PDB3)</td>
<td>iSCSI luns on ASA A400</td>
</tr>
</tbody>
</table>

Key factors for deployment consideration

- **Oracle database storage layout.** In this automated Oracle deployment, we provision four database volumes to host Oracle binary, data, and logs by default. We then create two ASM disk groups from data and logs luns. Within the +DATA asm disk group, we provision two data luns in a volume on each ASA A400 cluster node. Within the +LOGS asm disk group, we create two luns in a log volume on a single ASA A400 node. Multiple luns laid out within an ONTAP volume provides better performance in general.

- **Multiple DB servers deployment.** The automation solution can deploy an Oracle container database to multiple DB servers in a single Ansible playbook run. Regardless of the number of DB servers, the playbook execution remains the same. In the event of multi-DB server deployments, the playbook builds with an algorithm to place database luns on dual controllers of ASA A400 optimally. The binary and logs luns of odd number DB server in server hosts index place on controller 1. The binary and logs luns of even number DB server in the server hosts index place on controller 2. The DB data luns evenly distributed to two controllers. Oracle ASM combines the data luns on two controllers into a single ASM disk group to fully utilize the processing power of both controllers.

- **iSCSI configuration.** The database VMs connect to ASA storage with the iSCSI protocol for storage access. You should configure dual paths on each controller node for redundancy and set up iSCSI multi-path on the DB server for multi-path storage access. Enable jumbo frame on storage network to maximize performance and throughput.

- **Oracle ASM redundancy level to use for each Oracle ASM disk group that you create.** Because the ASA A400 configures storage in RAID DP for data protection at the cluster disk level, you should use **External Redundancy**, which means that the option does not allow Oracle ASM to mirror the contents of the disk group.

- **Database backup.** NetApp provides a SnapCenter software suite for database backup, restore, and cloning with a user-friendly UI interface. NetApp recommends implementing such a management tool to achieve fast (under a minute) SnapShot backup, quick (minutes) database restore, and database clone.

Solution deployment

The following sections provide step-by-step procedures for automated Oracle 19c deployment and protection in NetApp ASA A400 with directly mounted database luns via iSCSI to DB VM in a single node Restart configuration with Oracle ASM as database volume manager.

Prerequisites for deployment
Deployment requires the following prerequisites.

1. It is assumed that the NetApp ASA storage array has been installed and configured. This includes iSCSI broadcast domain, LACP interface groups a0a on both controller nodes, iSCSI VLAN ports (a0a-<iscsi-a-vlan-id>, a0a-<iscsi-b-vlan-id>) on both controller nodes. The following link provides detailed step-by-step instructions if help is needed. [Detailed guide - ASA A400](#)

2. Provision a Linux VM as an Ansible controller node with the latest version of Ansible and Git installed. Refer to the following link for details: [Getting Started with NetApp solution automation](#) in section - Setup the Ansible Control Node for CLI deployments on RHEL / CentOS or Setup the Ansible Control Node for CLI deployments on Ubuntu / Debian.

3. Clone a copy of the NetApp Oracle deployment automation toolkit for iSCSI.

   ```
   git clone https://bitbucket.ngage.netapp.com/scm/ns-bb/na_oracle_deploy_iscsi.git
   ```

4. Provision a Windows server to run the NetApp SnapCenter UI tool with the latest version. Refer to the following link for details: [Install the SnapCenter Server](#)

5. Build two RHEL Oracle DB servers either bare metal or virtualized VM. Create an admin user on DB servers with sudo without password privilege and enable SSH private/public key authentication between Ansible host and Oracle DB server hosts. Stage following Oracle 19c installation files on DB servers /tmp/archive directory.

   ```
   installer_archives:
   - "LINUX.X64_193000_grid_home.zip"
   - "p34762026_190000_Linux-x86-64.zip"
   - "LINUX.X64_193000_db_home.zip"
   - "p34765931_190000_Linux-x86-64.zip"
   - "p6880880_190000_Linux-x86-64.zip"
   ```

   Ensure that you have allocated at least 50G in Oracle VM root volume to have sufficient space to stage Oracle installation files.

6. Watch the following video:

   [Simplified and automated Oracle deployment on NetApp ASA with iSCSI](#)

**Automation parameter files**
Ansible playbook executes database installation and configuration tasks with predefined parameters. For this Oracle automation solution, there are three user-defined parameter files that need user input before playbook execution.

- hosts - define targets that the automation playbook is running against.
- vars/vars.yml - the global variable file that defines variables that apply to all targets.
- host_vars/host_name.yml - the local variable file that defines variables that apply only to a local target. In our use case, these are the Oracle DB servers.

In addition to these user-defined variable files, there are several default variable files that contain default parameters that do not require change unless necessary. The following sections show how the user-defined variable files are configured.

Parameter files configuration
1. Ansible target hosts file configuration:

```yaml
# Enter NetApp ASA controller management IP address
[ontap]
172.16.9.32

# Enter Oracle servers names to be deployed one by one, follow by each Oracle server public IP address, and ssh private key of admin user for the server.
[oracle]
ora_01 ansible_host=10.61.180.21 ansible_ssh_private_key_file =ora_01.pem
ora_02 ansible_host=10.61.180.23 ansible_ssh_private_key_file =ora_02.pem
```

2. Global vars/vars.yml file configuration

```yaml
# Enter the supported ONTAP platform: on-prem, aws-fsx.
ontap_platform: on-prem

# Enter ONTAP cluster management user credentials
username: "xxxxxxxx"
password: "xxxxxxxx"

# Enter Oracle SVM iSCSI lif addresses. Each controller configures
```
with dual paths iscsi_a, iscsi_b for redundancy

ora_iscsi_lif_mgmt:
  - {name: '{{ svm_name }}_mgmt', address: 172.21.253.220, netmask: 255.255.255.0, vlan_name: ora_mgmt, vlan_id: 3509}

ora_iscsi_lifs_node1:
  - {name: '{{ svm_name }}_lif_1a', address: 172.21.234.221, netmask: 255.255.255.0, vlan_name: ora_iscsi_a, vlan_id: 3490}
  - {name: '{{ svm_name }}_lif_1b', address: 172.21.235.221, netmask: 255.255.255.0, vlan_name: ora_iscsi_b, vlan_id: 3491}

ora_iscsi_lifs_node2:
  - {name: '{{ svm_name }}_lif_2a', address: 172.21.234.223, netmask: 255.255.255.0, vlan_name: ora_iscsi_a, vlan_id: 3490}
  - {name: '{{ svm_name }}_lif_2b', address: 172.21.235.223, netmask: 255.255.255.0, vlan_name: ora_iscsi_b, vlan_id: 3491}

###                   Linux env specific config variables
###                   Oracle DB env specific config variables

# Enter RHEL subscription to enable repo
redhat_sub_username: xxxxxxxx
redhat_sub_password: "xxxxxxxx"

# Enter Database domain name
db_domain: solutions.netapp.com

# Enter initial password for all required Oracle passwords. Change them after installation.
initial_pwd_all: xxxxxxxx

3. Local DB server host_vars/host_name.yml configuration
# User configurable Oracle host specific parameters

# Enter container database SID. By default, a container DB is created with 3 PDBs within the CDB
**oracle_sid**: NTAP1

# Enter database shared memory size or SGA. CDB is created with SGA at 75% of **memory_limit**, MB. The grand total of SGA should not exceed 75% available RAM on node.
**memory_limit**: 8192

Playbook execution
There are a total of six playbooks in the automation toolkit. Each performs different task blocks and serves different purposes.

0-all_playbook.yml - execute playbooks from 1-4 in one playbook run.
1-ansible_requirements.yml - set up Ansible controller with required libs and collections.
2-linux_config.yml - execute Linux kernel configuration on Oracle DB servers.
3-ontap_config.yml - configure ONTAP svm/volumes/luns for Oracle database and grant DB server access to luns.
4-oracle_config.yml - install and configure Oracle on DB servers for grid infrastructure and create a container database.
5-destroy.yml - optional to undo the environment to dismantle all.

There are three options to run the playbooks with the following commands.

1. Execute all deployment playbooks in one combined run.

   ansible-playbook -i hosts 0-all_playbook.yml -u admin -e @vars/vars.yml

2. Execute playbooks one at a time with the number sequence from 1-4.

   ansible-playbook -i hosts 1-ansible_requirements.yml -u admin -e @vars/vars.yml

   ansible-playbook -i hosts 2-linux_config.yml -u admin -e @vars/vars.yml

   ansible-playbook -i hosts 3-ontap_config.yml -u admin -e @vars/vars.yml

   ansible-playbook -i hosts 4-oracle_config.yml -u admin -e @vars/vars.yml

3. Execute 0-all_playbook.yml with a tag.
ansible-playbook -i hosts 0-all_playbook.yml -u admin -e @vars/vars.yml -t ansible_requirements

ansible-playbook -i hosts 0-all_playbook.yml -u admin -e @vars/vars.yml -t linux_config

ansible-playbook -i hosts 0-all_playbook.yml -u admin -e @vars/vars.yml -t ontap_config

ansible-playbook -i hosts 0-all_playbook.yml -u admin -e @vars/vars.yml -t oracle_config

4. Undo the environment

ansible-playbook -i hosts 5-destroy.yml -u admin -e @vars/vars.yml

Post execution validation
After the playbook run, login to the Oracle DB server as oracle user to validate that Oracle grid infrastructure and database are created successfully. Following is an example of Oracle database validation on host ora_01.

1. Validate the grid infrastructure and resources created.

```bash
[oracle@ora_01 ~]$ df -h
Filesystem           Size  Used Avail Use% Mounted on
/devtmpfs            7.7G   40K  7.7G   1% /dev
tmpfs                 7.8G  1.1G  6.7G  15% /dev/shm
tmpfs                 7.8G  312M  7.5G   4% /run
tmpfs                 7.8G     0  7.8G   0% /sys/fs/cgroup
/dev/mapper/rhel-root  44G   38G  6.8G  85% /
/dev/sdal            1014M  258M  757M  26% /boot
tmpfs                 1.6G  12K  1.6G   1% /run/user/42
tmpfs                 1.6G  4.0K  1.6G   1% /run/user/1000
/dev/mapper/ora_01_biny_01p1  40G   21G   20G  52% /u01
[oracle@ora_01 ~]$ asm
[oracle@ora_01 ~]$ crsctl stat res -t
```

```
Name            Target  State        Server   State
details

Local Resources

ora.DATA.dg
  ONLINE  ONLINE  ora_01  STABLE
ora.LISTENER.lsnr
  ONLINE  INTERMEDIATE  ora_01  Not All
Endpoints Re
  gistered,STABLE
ora.LOGS.dg
  ONLINE  ONLINE  ora_01  STABLE
ora.asm
  ONLINE  ONLINE  ora_01
Started,STABLE
ora.ons
  OFFLINE  OFFLINE  ora_01  STABLE

Cluster Resources
```
ora.cssd
  1        ONLINE  ONLINE       ora_01                 STABLE
ora.diskmon
  1        OFFLINE OFFLINE                               STABLE
ora.driver.afd
  1        ONLINE  ONLINE       ora_01                 STABLE
ora.evmd
  1        ONLINE  ONLINE       ora_01                 STABLE
ora.ntap1.db
  1        ONLINE  ONLINE       ora_01
Open,HOME=/u01/app/oracle/product/19.0.0
/NTAP1,STABLE

-----------------------------
[oracle@ora_01 ~]$ 

Ignore the Not All Endpoints Registered in State details. This results from a conflict of manual and dynamic database registration with the listener and can be safely ignored.

2. Validate ASM filter driver is working as expected.
<table>
<thead>
<tr>
<th>State</th>
<th>Type</th>
<th>Rebal</th>
<th>Sector</th>
<th>Logical_Sector</th>
<th>Block</th>
<th>AU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOUNTED</td>
<td>EXTERN</td>
<td>N</td>
<td>512</td>
<td>512</td>
<td>4096</td>
<td>4194304</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>327680</td>
<td>318644</td>
<td>0</td>
<td>318644</td>
</tr>
<tr>
<td>N</td>
<td>DATA/</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOUNTED</td>
<td>EXTERN</td>
<td>N</td>
<td>512</td>
<td>512</td>
<td>4096</td>
<td>4194304</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>81920</td>
<td>78880</td>
<td>0</td>
<td>78880</td>
</tr>
<tr>
<td>N</td>
<td>LOGS/</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ASMCMD> lsdsk
Path
AFD:ORA_01_DAT1_01
AFD:ORA_01_DAT1_03
AFD:ORA_01_DAT1_05
AFD:ORA_01_DAT1_07
AFD:ORA_01_DAT2_02
AFD:ORA_01_DAT2_04
AFD:ORA_01_DAT2_06
AFD:ORA_01_DAT2_08
AFD:ORA_01_LOGS_01
AFD:ORA_01_LOGS_02

ASMCMD> afd_state
ASMCMD-9526: The AFD state is 'LOADED' and filtering is 'ENABLED' on host 'ora_01'

3. Login to Oracle Enterprise Manager Express to validate database.
Enable additional port from sqlplus for login to individual container database or PDBs.

SQL> show pdbs

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>CON_NAME</th>
<th>OPEN MODE</th>
<th>RESTRICTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>READ ONLY</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>NTAP1_PDB1</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>4</td>
<td>NTAP1_PDB2</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>5</td>
<td>NTAP1_PDB3</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
</tbody>
</table>

SQL> alter session set container=NTAP1_PDB1;
Session altered.

SQL> select dbms_xdb_config.gethttpsport() from dual;

DBMS_XDB_CONFIG.GETHTTPSPORT()
-------------------------------
0

SQL> exec DBMS_XDB_CONFIG.SETHTTPSPORT(5501);
PL/SQL procedure successfully completed.

SQL> select dbms_xdb_config.gethttpsport() from dual;

DBMS_XDB_CONFIG.GETHTTPSPORT()
-------------------------------
5501
login to NTAP1_PDB1 from port 5501.
Oracle backup, restore, and clone with SnapCenter

Refer to TR-4979 Simplified, self-managed Oracle in VMware Cloud on AWS with guest-mounted FSx ONTAP section Oracle backup, restore, and clone with SnapCenter for details on setting up SnapCenter and executing the database backup, restore, and clone workflows.

Where to find additional information

To learn more about the information described in this document, review the following documents and/or websites:

- NETAPP ASA: ALL-FLASH SAN ARRAY
  https://www.netapp.com/data-storage/all-flash-san-storage-array/

- Installing Oracle Grid Infrastructure for a Standalone Server with a New Database Installation

- Installing and Configuring Oracle Database Using Response Files

- Use Red Hat Enterprise Linux 8.2 with ONTAP
  https://docs.netapp.com/us-en/ontap-sanhost/hu_rhel_82.html#all-san-array-configurations
TR-4979: Simplified, Self-managed Oracle in VMware Cloud on AWS with guest-mounted FSx ONTAP

Allen Cao, Niyaz Mohamed, NetApp

Purpose

Enterprises have been running Oracle on VMware in private data centers for decades. VMware Cloud (VMC) on AWS provides a push-button solution to bring VMware’s enterprise-class Software-Defined Data Center (SDDC) software to the AWS Cloud’s dedicated, elastic, bare-metal infrastructure. AWS FSx ONTAP offers premium storage to VMC SDDC and a data fabric that enables customers to run business-critical applications such as Oracle across vSphere®-based private, public, and hybrid cloud environments, with optimized access to AWS services. Whether it is an existing or new Oracle workload, VMC on AWS provides a familiar, simplified, and self-managed Oracle environment on VMware with all the benefits of AWS cloud while deferring all platform management and optimization to VMware.

This documentation demonstrates the deployment and protection of an Oracle database in a VMC environment with Amazon FSx ONTAP as primary database storage. Oracle database can be deployed to VMC on FSx storage as direct VM guest-mounted LUNs or NFS-mounted VMware VMDK datastore disks. This technical report focuses on Oracle database deployment as direct guest-mounted FSx storage to VMs in the VMC cluster with the iSCSI protocol and Oracle ASM. We also demonstrate how to use the NetApp SnapCenter UI tool to backup, restore, and clone an Oracle database for dev/test or other use cases for storage-efficient database operation in the VMC on AWS.

This solution addresses the following use cases:

- Oracle database deployment in VMC on AWS with Amazon FSx ONTAP as primary database storage
- Oracle database backup and restore in VMC on AWS using NetApp SnapCenter tool
- Oracle database clone for dev/test or other use cases in VMC on AWS using NetApp SnapCenter tool

Audience

This solution is intended for the following people:

- A DBA who would like to deploy Oracle in VMC on AWS with Amazon FSx ONTAP
- A database solution architect who would like to test Oracle workloads in VMC on the AWS cloud
- A storage administrator who would like to deploy and manage an Oracle database deployed to VMC on AWS with Amazon FSx ONTAP
- An application owner who would like to stand up an Oracle database in VMC on the AWS cloud

Solution test and validation environment

The testing and validation of this solution was performed in a lab environment with VMC on AWS that might not match the final deployment environment. For more information, see the section [Key Factors for Deployment Consideration].

Architecture
## Hardware and software components

### Hardware

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSx ONTAP storage</td>
<td>Current version offered by AWS</td>
</tr>
<tr>
<td>One FSx ONTAP HA cluster</td>
<td>In the same VPC and availability zone as VMC</td>
</tr>
<tr>
<td>VMC SDDC cluster</td>
<td>Amazon EC2 i3.metal single node/Intel Xeon E5-2686 CPU, 36 cores/512G RAM</td>
</tr>
<tr>
<td></td>
<td>10.37 TB vSAN storage</td>
</tr>
</tbody>
</table>

### Software

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RedHat Linux</td>
<td>RHEL-8.6, 4.18.0-372.9.1.el8.x86_64 kernel</td>
</tr>
<tr>
<td></td>
<td>Deployed RedHat subscription for testing</td>
</tr>
<tr>
<td>Windows Server</td>
<td>2022 Standard, 10.0.20348 Build 20348</td>
</tr>
<tr>
<td></td>
<td>Hosting SnapCenter server</td>
</tr>
<tr>
<td>Oracle Grid Infrastructure</td>
<td>Version 19.18</td>
</tr>
<tr>
<td></td>
<td>Applied RU patch p34762026_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle Database</td>
<td>Version 19.18</td>
</tr>
<tr>
<td></td>
<td>Applied RU patch p34765931_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle OPatch</td>
<td>Version 12.2.0.1.36</td>
</tr>
<tr>
<td></td>
<td>Latest patch p6880880_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>SnapCenter Server</td>
<td>Version 4.9P1</td>
</tr>
<tr>
<td></td>
<td>Workgroup deployment</td>
</tr>
</tbody>
</table>
BlueXP backup and recovery for VMs

VMware vSphere

Open JDK

Release 1.0

Version 8.0.1.00300

Version java-1.8.0-openjdk.x86_64

Deployed as an ova vSphere plugin VM

VMware Tools, Version: 11365 - Linux, 12352 - Windows

SnapCenter plugin requirement on DB VMs

Oracle database configuration in VMC on AWS

<table>
<thead>
<tr>
<th>Server</th>
<th>Database</th>
<th>DB Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora_01</td>
<td>cdb1(cdb1_pdb1,cdb1_pdb2,cdb1_pdb3)</td>
<td>VMDK datastore on FSx ONTAP</td>
</tr>
<tr>
<td>ora_01</td>
<td>cdb2(cdb2_pdb)</td>
<td>VMDK datastore on FSx ONTAP</td>
</tr>
<tr>
<td>ora_02</td>
<td>cdb3(cdb3_pdb1,cdb3_pdb2,cdb3_pdb3)</td>
<td>Direct guest mounted FSx ONTAP</td>
</tr>
<tr>
<td>ora_02</td>
<td>cdb4(cdb4_pdb)</td>
<td>Direct guest mounted FSx ONTAP</td>
</tr>
</tbody>
</table>

Key factors for deployment consideration

- **FSx to VMC connectivity.** When you deploy your SDDC on VMware Cloud on AWS, it is created within an AWS account and a VPC dedicated to your organization and managed by VMware. You must also connect the SDDC to an AWS account belonging to you, called the customer AWS account. This connection allows your SDDC to access AWS services belonging to your customer account. FSx for ONTAP is an AWS service deployed in your customer account. Once the VMC SDDC is connected to your customer account, FSx storage is available to VMs in VMC SDDC for direct guest mount.

- **FSx storage HA clusters single- or multi-zone deployment.** In these tests and validations, we deployed an FSx HA cluster in a single AWS availability zone. NetApp also recommends deploying FSx for NetApp ONTAP and VMware Cloud on AWS in the same availability zone to achieve better performance and avoid data transfer charges between availability zones.

- **FSx storage cluster sizing.** An Amazon FSx for ONTAP storage file system provides up to 160,000 raw SSD IOPS, up to 4GBps throughput, and a maximum of 192TiB capacity. However, you can size the cluster in terms of provisioned IOPS, throughput, and storage limit (minimum 1,024 GiB) based on your actual requirements at the time of deployment. The capacity can be adjusted dynamically on the fly without affecting application availability.

- **Oracle data and logs layout.** In our tests and validations, we deployed two ASM disk groups for data and logs respectively. Within the +DATA asm disk group, we provisioned four LUNs in a data volume. Within the +LOGS asm disk group, we provisioned two LUNs in a log volume. In general, multiple LUNs laid out within an Amazon FSx for ONTAP volume provide better performance.

- **iSCSI configuration.** The database VMs in VMC SDDC connect to FSx storage with the iSCSI protocol. It is important to gauge the Oracle database peak I/O throughput requirement by carefully analyzing the Oracle AWR report to determine the application and iSCSI traffic-throughput requirements. NetApp also recommends allocating four iSCSI connections to both FSx iSCSI endpoints with multipath properly configured.

- **Oracle ASM redundancy level to use for each Oracle ASM disk group that you create.** Because FSx ONTAP already mirrors the storage on the FSx cluster level, you should use External Redundancy, which means that the option does not allow Oracle ASM to mirror the contents of the disk group.
• **Database backup.** NetApp provides a SnapCenter software suite for database backup, restore, and cloning with a user-friendly UI interface. NetApp recommends implementing such a management tool to achieve fast (under a minute) SnapShot backup, quick (minutes) database restore, and database clone.

**Solution deployment**

The following sections provide step-by-step procedures for Oracle 19c deployment in VMC on AWS with directly mounted FSx ONTAP storage to DB VM in a single node Restart configuration with Oracle ASM as database volume manager.

**Prerequisites for deployment**
Deployment requires the following prerequisites.

1. A software-defined data center (SDDC) using VMware Cloud on AWS has been created. For detailed instruction on how to create an SDDC in VMC, please refer to VMware documentation Getting Started With VMware Cloud on AWS.

2. An AWS account has been set up, and the necessary VPC and network segments have been created within your AWS account. The AWS account is linked to your VMC SDDC.

3. From the AWS EC2 console, deploying an Amazon FSx for ONTAP storage HA clusters to host the Oracle database volumes. If you are not familiar with the deployment of FSx storage, see the documentation Creating FSx for ONTAP file systems for step-by-step instructions.

4. The above step can be performed using the following Terraform automation toolkit, which creates an EC2 instance as a jump host for SDDC in VMC access via SSH and an FSx file system. Review instructions carefully and change the variables to suit your environment before execution.

   ```
   git clone https://github.com/NetApp-Automation/na_aws_fsx_ec2_deploy.git
   ```

5. Build VMs in VMware SDDC on AWS for hosting your Oracle environment to be deployed in VMC. In our demonstration, we have built two Linux VMs as Oracle DB servers, one Windows server for the SnapCenter server, and one optional Linux server as an Ansible controller for automated Oracle installation or configuration if desired. Following is a snapshot of the lab environment for the solution validation.

6. Optionally, NetApp also provides several automation toolkits to run Oracle deployment and configuration when applicable. Refer to DB Automation Toolkits for more information.

   Ensure that you have allocated at least 50G in Oracle VM root volume in order to have sufficient space to stage Oracle installation files.
DB VM kernel configuration
With the prerequisites provisioned, login to the Oracle VM as an admin user via SSH and sudo to the root user to configure the Linux kernel for Oracle installation. Oracle install files can be staged in an AWS S3 bucket and transferred into the VM.

1. Create a staging directory `/tmp/archive` folder and set the 777 permission.

```
mkdir /tmp/archive
chmod 777 /tmp/archive
```

2. Download and stage the Oracle binary installation files and other required rpm files to the `/tmp/archive` directory.

See the following list of installation files to be stated in `/tmp/archive` on the DB VM.

```
[admin@ora_02 ~]$ ls -l /tmp/archive/
total 10539364
-rw-rw-r--. 1 admin  admin         19112 Oct  4 17:04 compat-libcap1-1.10-7.el7.x86_64.rpm
-rw-rw-r--. 1 admin  admin    3059705302 Oct  4 17:10 LINUX.X64_193000_db_home.zip
-rw-rw-r--. 1 admin  admin    2889184573 Oct  4 17:11 LINUX.X64_193000_grid_home.zip
-rw-rw-r--. 1 admin  admin    2872741741 Oct  4 17:12 p34762026_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    1843577895 Oct  4 17:13 p34765931_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin     124347218 Oct  4 17:13 p6880880_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    2872741741 Oct  4 17:12 p34762026_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    2889184573 Oct  4 17:11 LINUX.X64_193000_grid_home.zip
-rw-rw-r--. 1 admin  admin    1843577895 Oct  4 17:13 p34765931_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    2872741741 Oct  4 17:12 p34762026_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    1843577895 Oct  4 17:13 p34765931_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    2872741741 Oct  4 17:12 p34762026_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    1843577895 Oct  4 17:13 p34765931_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    2872741741 Oct  4 17:12 p34762026_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    1843577895 Oct  4 17:13 p34765931_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    2872741741 Oct  4 17:12 p34762026_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    1843577895 Oct  4 17:13 p34765931_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    2872741741 Oct  4 17:12 p34762026_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    1843577895 Oct  4 17:13 p34765931_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    2872741741 Oct  4 17:12 p34762026_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    1843577895 Oct  4 17:13 p34765931_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    2872741741 Oct  4 17:12 p34762026_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    1843577895 Oct  4 17:13 p34765931_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    2872741741 Oct  4 17:12 p34762026_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    1843577895 Oct  4 17:13 p34765931_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    2872741741 Oct  4 17:12 p34762026_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    1843577895 Oct  4 17:13 p34765931_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    2872741741 Oct  4 17:12 p34762026_190000_Linux-x86-64.zip
-rw-rw-r--. 1 admin  admin    1843577895 Oct  4 17:13 p34765931_190000_Linux-x86-64.zip
```

3. Install Oracle 19c preinstall RPM, which satisfies most kernel configuration requirements.

```
yum install /tmp/archive/oracle-database-preinstall-19c-1.0-2.el8.x86_64.rpm
```
4. Download and install the missing compat-libcap1 in Linux 8.

```
yum install /tmp/archive/compat-libcap1-1.10-7.el7.x86_64.rpm
```

5. From NetApp, download and install NetApp host utilities.

```
yum install /tmp/archive/netapp_linux_unified_host_utilities-7-1.x86_64.rpm
```


```
yum install /tmp/archive/policycoreutils-python-utils-2.9-9.el8.noarch.rpm
```

7. Install open JDK version 1.8.

```
yum install java-1.8.0-openjdk.x86_64
```

8. Install iSCSI initiator utils.

```
yum install iscsi-initiator-utils
```

9. Install sg3_utils.

```
yum install sg3_utils
```

10. Install device-mapper-multipath.

```
yum install device-mapper-multipath
```

11. Disable transparent hugepages in the current system.

```
echo never > /sys/kernel/mm/transparent_hugepage/enabled
```

```
echo never > /sys/kernel/mm/transparent_hugepage/defrag
```

12. Add the following lines in `/etc/rc.local` to disable transparent HugePage after reboot.
# Disable transparent hugepages
if test -f /sys/kernel/mm/transparent_hugepage/enabled;
then
    echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi
if test -f /sys/kernel/mm/transparent_hugepage/defrag;
then
    echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi

13. Disable selinux by changing SELINUX=enforcing to SELINUX=disabled. You must reboot the host to make the change effective.

vi /etc/sysconfig/selinux

14. Add the following lines to limit.conf to set the file descriptor limit and stack size.

vi /etc/security/limits.conf

* hard nofile 65536
* soft stack 10240

15. Add swap space to DB VM if there is no swap space configured with this instruction: How do I allocate memory to work as swap space in an Amazon EC2 instance by using a swap file? The exact amount of space to add depends on the size of RAM up to 16G.

16. Change node.session.timeo.replacement_timeout in the iscsi.conf configuration file from 120 to 5 seconds.

vi /etc/iscsi/iscsid.conf

17. Enable and start the iSCSI service on the EC2 instance.

systemctl enable iscsid

systemctl start iscsid
18. Retrieve the iSCSI initiator address to be used for database LUN mapping.

```bash
cat /etc/iscsi/initiatorname.iscsi
```

19. Add the asm groups for asm management user (oracle).

```bash
groupadd asmadmin

groupadd asmdba

groupadd asmoper
```

20. Modify the oracle user to add asm groups as secondary groups (the oracle user should have been created after Oracle preinstall RPM installation).

```bash
usermod -a -G asmadmin oracle

usermod -a -G asmdba oracle

usermod -a -G asmoper oracle
```

21. Stop and disable the Linux firewall if it is active.

```bash
systemctl stop firewalld

systemctl disable firewalld
```

22. Enable password-less sudo for admin user by uncommenting # %wheel ALL=(ALL) NOPASSWD: ALL line in /etc/sudoers file. Change the file permission to make the edit.

```bash
chmod 640 /etc/sudoers

vi /etc/sudoers
```
chmod 440 /etc/sudoers

23. Reboot the EC2 instance.

Provision and map FSx ONTAP LUNs to the DB VM
Provision three volumes from the command line by login to FSx cluster as fsxadmin user via ssh and FSx cluster management IP. Create LUNs within the volumes to host the Oracle database binary, data, and logs files.

1. Log into the FSx cluster through SSH as the fsxadmin user.

```bash
ssh fsxadmin@10.49.0.74
```

2. Execute the following command to create a volume for the Oracle binary.

```bash
vol create -volume ora_02_biny -aggregate aggr1 -size 50G -state online -type RW -snapshot-policy none -tiering-policy snapshot-only
```

3. Execute the following command to create a volume for Oracle data.

```bash
vol create -volume ora_02_data -aggregate aggr1 -size 100G -state online -type RW -snapshot-policy none -tiering-policy snapshot-only
```

4. Execute the following command to create a volume for Oracle logs.

```bash
vol create -volume ora_02_logs -aggregate aggr1 -size 100G -state online -type RW -snapshot-policy none -tiering-policy snapshot-only
```

5. Validate the volumes created.

```bash
vol show ora*
```

Output from the command:

```
FsxId0c00cec8dad373fd1::> vol show ora*
Vserver  Volume   Aggregate    State      Type       Size
Available Used%
--------- ------------ ----------- ------- ---- ----------
---------- ----- 

nim       ora_02_biny  aggr1        online     RW         50GB
22.98GB   51%
nim       ora_02_data  aggr1        online     RW        100GB
18.53GB   80%
nim       ora_02_logs  aggr1        online     RW         50GB
 7.98GB  83%
```
6. Create a binary LUN within the database binary volume.

```bash
lun create -path /vol/ora_02_biny/ora_02_biny_01 -size 40G -ostype linux
```

7. Create data LUNs within the database data volume.

```bash
lun create -path /vol/ora_02_data/ora_02_data_01 -size 20G -ostype linux
lun create -path /vol/ora_02_data/ora_02_data_02 -size 20G -ostype linux
lun create -path /vol/ora_02_data/ora_02_data_03 -size 20G -ostype linux
lun create -path /vol/ora_02_data/ora_02_data_04 -size 20G -ostype linux
```

8. Create log LUNs within the database logs volume.

```bash
lun create -path /vol/ora_02_logs/ora_02_logs_01 -size 40G -ostype linux
lun create -path /vol/ora_02_logs/ora_02_logs_02 -size 40G -ostype linux
```

9. Create an igroup for the EC2 instance with the initiator retrieved from step 14 of the EC2 kernel configuration above.

```bash
igroup create -igroup ora_02 -protocol iscsi -ostype linux
```

10. Map the LUNs to the igroup created above. Increment the LUN ID sequentially for each additional LUN.
lun map -path /vol/ora_02_biny/ora_02_biny_01 -igroup ora_02
-vserver svm ora -lun-id 0
lun map -path /vol/ora_02_data/ora_02_data_01 -igroup ora_02
-vserver svm ora -lun-id 1
lun map -path /vol/ora_02_data/ora_02_data_02 -igroup ora_02
-vserver svm ora -lun-id 2
lun map -path /vol/ora_02_data/ora_02_data_03 -igroup ora_02
-vserver svm ora -lun-id 3
lun map -path /vol/ora_02_data/ora_02_data_04 -igroup ora_02
-vserver svm ora -lun-id 4
lun map -path /vol/ora_02_logs/ora_02_logs_01 -igroup ora_02
-vserver svm ora -lun-id 5
lun map -path /vol/ora_02_logs/ora_02_logs_02 -igroup ora_02
-vserver svm ora -lun-id 6

11. Validate the LUN mapping.

mapping show

This is expected to return:

FsxId0c00cec8dad373fd1::> mapping show
(lun mapping show)

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Path</th>
<th>Igroup</th>
<th>LUN ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>/vol/ora_02_biny/ora_02_u01_01</td>
<td>ora_02</td>
<td>0</td>
</tr>
<tr>
<td>iscsi</td>
<td>/vol/ora_02_data/ora_02_u02_01</td>
<td>ora_02</td>
<td>1</td>
</tr>
<tr>
<td>iscsi</td>
<td>/vol/ora_02_data/ora_02_u02_02</td>
<td>ora_02</td>
<td>2</td>
</tr>
<tr>
<td>iscsi</td>
<td>/vol/ora_02_data/ora_02_u02_03</td>
<td>ora_02</td>
<td>3</td>
</tr>
<tr>
<td>iscsi</td>
<td>/vol/ora_02_data/ora_02_u02_04</td>
<td>ora_02</td>
<td>4</td>
</tr>
<tr>
<td>iscsi</td>
<td>/vol/ora_02_logs/ora_02_u03_01</td>
<td>ora_02</td>
<td>5</td>
</tr>
<tr>
<td>iscsi</td>
<td>/vol/ora_02_logs/ora_02_u03_02</td>
<td>ora_02</td>
<td>6</td>
</tr>
</tbody>
</table>
DB VM storage configuration
Now, import and set up the FSx ONTAP storage for the Oracle grid infrastructure and database installation on the VMC database VM.

1. Login to the DB VM via SSH as the admin user using Putty from Windows jump server.
2. Discover the FSx iSCSI endpoints using either SVM iSCSI IP address. Change to your environment-specific portal address.

```bash
sudo iscsiadm --mode discovery --op update --type sendtargets --portal 10.49.0.12
```

3. Establish iSCSI sessions by logging into each target.

```bash
sudo iscsiadm --mode node -l all
```

The expected output from the command is:

```
[ec2-user@ip-172-30-15-58 ~]$ sudo iscsiadm --mode node -l all
```

4. View and validate a list of active iSCSI sessions.

```bash
sudo iscsiadm --mode session
```

Return the iSCSI sessions.

```
[ec2-user@ip-172-30-15-58 ~]$ sudo iscsiadm --mode session
tcp: [1] 10.49.0.186:3260,1028 iqn.1992-08.com.netapp:sn.545a38bf06ac11ee8503e395ab90d704:vs.3 (non-flash)
```
5. Verify that the LUNs were imported into the host.

```
sudo sanlun lun show
```

This will return a list of Oracle LUNs from FSx.

```
[admin@ora_02 ~]$ sudo sanlun lun show
calendar(7mode/E-Series)/
device          host                  lun
vserver(cDOT/FlashRay)        lun-pathname
filename        adapter    protocol   size    product
--------------------------------------------------------------------
--------------------------------------------------------------------
nim                           /vol/ora_02_logs/ora_02_u03_02
/dev/sdo        host34     iSCSI      20g     cDOT
nim                           /vol/ora_02_logs/ora_02_u03_01
/dev/sdn        host34     iSCSI      20g     cDOT
nim                           /vol/ora_02_data/ora_02_u02_04
/dev/sdm        host34     iSCSI      20g     cDOT
nim                           /vol/ora_02_data/ora_02_u02_03
/dev/sdl        host34     iSCSI      20g     cDOT
nim                           /vol/ora_02_data/ora_02_u02_02
/dev/sdk        host34     iSCSI      20g     cDOT
nim                           /vol/ora_02_data/ora_02_u02_01
/dev/sdj        host34     iSCSI      20g     cDOT
nim                           /vol/ora_02_biny/ora_02_u01_01
/dev/sdi        host34     iSCSI      40g     cDOT
nim                           /vol/ora_02_logs/ora_02_u03_02
/dev/sdh        host33     iSCSI      20g     cDOT
nim                           /vol/ora_02_logs/ora_02_u03_01
/dev/sdg        host33     iSCSI      20g     cDOT
nim                           /vol/ora_02_data/ora_02_u02_04
/dev/sdf        host33     iSCSI      20g     cDOT
nim                           /vol/ora_02_data/ora_02_u02_03
/dev/sde        host33     iSCSI      20g     cDOT
nim                           /vol/ora_02_data/ora_02_u02_02
/dev/sdd        host33     iSCSI      20g     cDOT
nim                           /vol/ora_02_data/ora_02_u02_01
/dev/sdc        host33     iSCSI      20g     cDOT
nim                           /vol/ora_02_biny/ora_02_u01_01
/dev/sdb        host33     iSCSI      40g     cDOT
```

6. Configure the `multipath.conf` file with following default and blacklist entries.
sudo vi /etc/multipath.conf

Add following entries:

defaults {
    find_multipaths yes
    user_friendly_names yes
}

blacklist {
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
    devnode "^hd[a-z]"
    devnode "^cciss.*"
}

7. Start the multipath service.

sudo systemctl start multipathd

Now multipath devices appear in the /dev/mapper directory.

[ec2-user@ip-172-30-15-58 ~]$ ls -l /dev/mapper
total 0
lrwxrwxrwx 1 root root 7 Mar 21 20:13 3600a0980c574235472455534e68512d -> ../dm-0
lrwxrwxrwx 1 root root 7 Mar 21 20:13 3600a0980c574235472455534e685141 -> ../dm-1
lrwxrwxrwx 1 root root 7 Mar 21 20:13 3600a0980c574235472455534e685142 -> ../dm-2
lrwxrwxrwx 1 root root 7 Mar 21 20:13 3600a0980c574235472455534e685143 -> ../dm-3
lrwxrwxrwx 1 root root 7 Mar 21 20:13 3600a0980c574235472455534e685144 -> ../dm-4
lrwxrwxrwx 1 root root 7 Mar 21 20:13 3600a0980c574235472455534e685145 -> ../dm-5
lrwxrwxrwx 1 root root 7 Mar 21 20:13 3600a0980c574235472455534e685146 -> ../dm-6
crw------- 1 root root 10, 236 Mar 21 18:19 control

8. Log into the FSx ONTAP cluster as the fsxadmin user via SSH to retrieve the serial-hex number for each LUN starting with 6c574xx..., the HEX number starts with 3600a0980, which is the AWS vendor ID.
lun show -fields serial-hex

and return as follow:

```
FsxId02ad7bf3476b741df::> lun show -fields serial-hex
vserver path serial-hex
----------------- ------------------------
svm_ora /vol/ora_02_biny/ora_02_biny_01 6c574235472455534e68512d
svm_ora /vol/ora_02_data/ora_02_data_01 6c574235472455534e685141
svm_ora /vol/ora_02_data/ora_02_data_02 6c574235472455534e685142
svm_ora /vol/ora_02_data/ora_02_data_03 6c574235472455534e685143
svm_ora /vol/ora_02_data/ora_02_data_04 6c574235472455534e685144
svm_ora /vol/ora_02_logs/ora_02_logs_01 6c574235472455534e685145
svm_ora /vol/ora_02_logs/ora_02_logs_02 6c574235472455534e685146
7 entries were displayed.
```

9. Update the `/dev/multipath.conf` file to add a user-friendly name for the multipath device.

```
sudo vi /etc/multipath.conf
```

with following entries:
multipaths {
    multipath {
        wwid            3600a09806c574235472455534e68512d
        alias           ora_02_biny_01
    }
    multipath {
        wwid            3600a09806c574235472455534e685141
        alias           ora_02_data_01
    }
    multipath {
        wwid            3600a09806c574235472455534e685142
        alias           ora_02_data_02
    }
    multipath {
        wwid            3600a09806c574235472455534e685143
        alias           ora_02_data_03
    }
    multipath {
        wwid            3600a09806c574235472455534e685144
        alias           ora_02_data_04
    }
    multipath {
        wwid            3600a09806c574235472455534e685145
        alias           ora_02_logs_01
    }
    multipath {
        wwid            3600a09806c574235472455534e685146
        alias           ora_02_logs_02
    }
}

10. Reboot the multipath service to verify that the devices under /dev/mapper have changed to LUN names versus serial-hex IDs.

    sudo systemctl restart multipathd

Check /dev/mapper to return as following:
11. Partition the binary LUN with a single primary partition.

   ```bash
   sudo fdisk /dev/mapper/ora_02_biny_01
   ```

12. Format the partitioned binary LUN with an XFS file system.

   ```bash
   sudo mkfs.xfs /dev/mapper/ora_02_biny_01p1
   ```

13. Mount the binary LUN to /u01.

   ```bash
   sudo mkdir /u01
   sudo mount -t xfs /dev/mapper/ora_02_biny_01p1 /u01
   ```

14. Change /u01 mount point ownership to the oracle user and its associated primary group.

   ```bash
   sudo chown oracle:oinstall /u01
   ```

15. Find the UUI of the binary LUN.

   ```bash
   sudo blkid /dev/mapper/ora_02_biny_01p1
   ```
16. Add a mount point to /etc/fstab.

```
sudo vi /etc/fstab
```

Add the following line.

```
UUID=d89fb1c9-4f89-4de4-b4d9-17754036d11d   /u01    xfs
defaults,nofail 0       2
```

17. As the root user, add the udev rule for Oracle devices.

```
vi /etc/udev/rules.d/99-oracle-asmdevices.rules
```

Include following entries:

```
ENV{DM_NAME}=="ora*", GROUP="oinstall", OWNER="oracle",
MODE="660"
```

18. As the root user, reload the udev rules.

```
udisksd control --reload-rules
```

19. As the root user, trigger the udev rules.

```
udisksd trigger
```

20. As the root user, reload multipathd.

```
systemctl restart multipathd
```

21. Reboot the EC2 instance host.

Oracle grid infrastructure installation
1. Log into the DB VM as the admin user via SSH and enable password authentication by uncommenting `PasswordAuthentication yes` and then commenting out `PasswordAuthentication no`.

```
sudo vi /etc/ssh/sshd_config
```

2. Restart the `sshd` service.

```
sudo systemctl restart sshd
```

3. Reset the Oracle user password.

```
sudo passwd oracle
```

4. Log in as the Oracle Restart software owner user (oracle). Create an Oracle directory as follows:

```
mkdir -p /u01/app/oracle
mkdir -p /u01/app/oraInventory
```

5. Change the directory permission setting.

```
chmod -R 775 /u01/app
```

6. Create a grid home directory and change to it.

```
mkdir -p /u01/app/oracle/product/19.0.0/grid
```

```
cd /u01/app/oracle/product/19.0.0/grid
```

7. Unzip the grid installation files.

```
unzip -q /tmp/archive/LINUX.X64_193000_grid_home.zip
```

8. From grid home, delete the `OPatch` directory.
9. From grid home, unzip `p6880880_190000_Linux-x86-64.zip`.

   ```bash
   unzip -q /tmp/archive/p6880880_190000_Linux-x86-64.zip
   ```

10. From grid home, revise `cv/admin/cvu_config`, uncomment and replace
    `CV_ASSUME_DISTID=OEL5` with `CV_ASSUME_DISTID=OL7`.

    ```bash
    vi cv/admin/cvu_config
    ```

11. Prepare a `gridsetup.rsp` file for silent installation and place the rsp file in the `/tmp/archive` directory. The rsp file should cover sections A, B, and G with the following information:

    ```
    INVENTORY_LOCATION=/u01/app/oraInventory
    oracle.install.option=HA_CONFIG
    ORACLE_BASE=/u01/app/oracle
    oracle.install.asm.OSDBA=asmdba
    oracle.install.asm.OSOPER=asmoper
    oracle.install.asm.OSASM=asmadmin
    oracle.install.asm.SYSASMPassword="SetPWD"
    oracle.install.asm.diskGroup.name=DATA
    oracle.install.asm.diskGroup.redundancy=EXTERNAL
    oracle.install.asm.diskGroup.AUSize=4
    oracle.install.asm.diskGroup.disks=/dev/mapper/ora_02_data_01,/dev/mapper/ora_02_data_02,/dev/mapper/ora_02_data_03,/dev/mapper/ora_02_data_04
    oracle.install.asm.diskGroup.diskDiscoveryString=/dev/mapper/*
    oracle.install.asm.monitorPassword="SetPWD"
    oracle.install.asm.configureAFD=true
    ```

12. Log into the EC2 instance as the root user and set `ORACLE_HOME` and `ORACLE_BASE`.

    ```bash
    export ORACLE_HOME=/u01/app/oracle/product/19.0.0/
    export ORACLE_BASE=/tmp
    cd /u01/app/oracle/product/19.0.0/grid/bin
    ```
13. Initialize disk devices for use with the Oracle ASM filter driver.

```bash
./asmcmd afd_label DATA01 /dev/mapper/ora_02_data_01 --init
./asmcmd afd_label DATA02 /dev/mapper/ora_02_data_02 --init
./asmcmd afd_label DATA03 /dev/mapper/ora_02_data_03 --init
./asmcmd afd_label DATA04 /dev/mapper/ora_02_data_04 --init
./asmcmd afd_label LOGS01 /dev/mapper/ora_02_logs_01 --init
./asmcmd afd_label LOGS02 /dev/mapper/ora_02_logs_02 --init
```

14. Install `cvuqdisk-1.0.10-1.rpm`.

```bash
rpm -ivh /u01/app/oracle/product/19.0.0/grid/cv/rpm/cvuqdisk-1.0.10-1.rpm
```

15. Unset `$ORACLE_BASE`.

```bash
unset ORACLE_BASE
```

16. Log into the EC2 instance as the Oracle user and extract the patch in the `/tmp/archive` folder.

```bash
unzip -q /tmp/archive/p34762026_190000_Linux-x86-64.zip -d /tmp/archive
```

17. From grid home `/u01/app/oracle/product/19.0.0/grid` and as the oracle user, launch `gridSetup.sh` for grid infrastructure installation.

```bash
./gridSetup.sh -applyRU /tmp/archive/34762026/-silent -responseFile /tmp/archive/gridsetup.rsp
```

18. As root user, execute the following script(s):
19. As root user, reload the multipathd.

   systemctl restart multipathd

20. As the Oracle user, execute the following command to complete the configuration:

   /u01/app/oracle/product/19.0.0/grid/gridSetup.sh -executeConfigTools
   -responseFile /tmp/archive/gridsetup.rsp -silent

21. As the Oracle user, create the LOGS disk group.

   bin/asmca -silent -sysAsmPassword 'yourPWD' -asmsnmpPassword
   'yourPWD' -createDiskGroup -diskGroupName LOGS -disk 'AFD:LOGS*' -redundancy EXTERNAL -au_size 4

22. As the Oracle user, validate grid services after installation configuration.

   bin/crsctl stat res -t
Validating ASM filter driver status.
[oracle@ora_02 grid]$ export ORACLE_HOME=/u01/app/oracle/product/19.0.0/grid
[oracle@ora_02 grid]$ export ORACLE_SID=+ASM
[oracle@ora_02 grid]$ export PATH=$PATH:$ORACLE_HOME/bin
[oracle@ora_02 grid]$ asmcmd
ASMCMD> lsdg
State    Type    Rebal Sector Logical_Sector Block       AU
Total_MB  Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks
Voting_files Name
MOUNTED  EXTERN N         512             512   4096  4194304
81920    81780                0           81780              0
N  DATA/
MOUNTED  EXTERN N         512             512   4096  4194304
40960    40852                0           40852              0
N  LOGS/
ASMCMD> afd_state
ASMCMD-9526: The AFD state is 'LOADED' and filtering is 'ENABLED' on host 'ora_02'
ASMCMD> exit
[oracle@ora_02 grid]$

24. Validate HA service status.

[oracle@ora_02 bin]$ ./crsctl check has
CRS-4638: Oracle High Availability Services is online

Oracle database installation
1. Log in as the Oracle user and unset `$ORACLE_HOME` and `$ORACLE_SID` if it is set.

   ```bash
   unset ORACLE_HOME
   unset ORACLE_SID
   ```

2. Create the Oracle DB home directory and change the directory to it.

   ```bash
   mkdir /u01/app/oracle/product/19.0.0/cdb3
   cd /u01/app/oracle/product/19.0.0/cdb3
   ```

3. Unzip the Oracle DB installation files.

   ```bash
   unzip -q /tmp/archive/LINUX.X64_193000_db_home.zip
   ```

4. From the DB home, delete the `OPatch` directory.

   ```bash
   rm -rf OPatch
   ```

5. From DB home, unzip `p6880880_190000_Linux-x86-64.zip`.

   ```bash
   unzip -q /tmp/archive/p6880880_190000_Linux-x86-64.zip
   ```

6. From DB home, revise `cv/admin/cvu_config` and uncomment and replace `CV_ASSUME_DISTID=OEL5` with `CV_ASSUME_DISTID=OL7`.

   ```bash
   vi cv/admin/cvu_config
   ```

7. From the `/tmp/archive` directory, unpack the DB 19.18 RU patch.

   ```bash
   unzip -q /tmp/archive/p34765931_190000_Linux-x86-64.zip -d /tmp/archive
   ```

8. Prepare the DB silent install rsp file in `/tmp/archive/dbinstall.rsp` directory with the following values:
oracle.install.option=INSTALL_DB_SWONLY
UNIX_GROUP_NAME=oinstall
INVENTORY_LOCATION=/u01/app/oraInventory
ORACLE_HOME=/u01/app/oracle/product/19.0.0/cdb3
ORACLE_BASE=/u01/app/oracle
oracle.install.db.InstallEdition=EE
oracle.install.db.OSDBA_GROUP=dba
oracle.install.db.OSOPER_GROUP=oper
oracle.install.db.OSBACKUPDBA_GROUP=oper
oracle.install.db.OSDGDBA_GROUP=dba
oracle.install.db.OSKMDBA_GROUP=dba
oracle.install.db.OSRACDBA_GROUP=dba
oracle.install.db.rootconfig.executeRootScript=false

9. From cdb3 home /u01/app/oracle/product/19.0.0/cdb3, execute silent software-only DB installation.

  ./runInstaller -applyRU /tmp/archive/34765931/ -silent
  -ignorePrereqFailure -responseFile /tmp/archive/dbinstall.rsp

10. As root user, run the root.sh script after software-only installation.

    /u01/app/oracle/product/19.0.0/db1/root.sh

11. As oracle user, create the dbca.rsp file with the following entries:
12. As oracle user, launch DB creation with dbca.

```
bin/dbca -silent -createDatabase -responseFile /tmp/archive/dbca.rsp
```

output:
Prepare for db operation
7% complete
Registering database with Oracle Restart
11% complete
Copying database files
33% complete
Creating and starting Oracle instance
35% complete
38% complete
42% complete
45% complete
48% complete
Completing Database Creation
53% complete
55% complete
56% complete
Creating Pluggable Databases
60% complete
64% complete
69% complete
78% complete
Executing Post Configuration Actions
100% complete
Database creation complete. For details check the logfiles at:
   /u01/app/oracle/cfgtoollogs/dbca/cdb3.
Database Information:
Global Database Name: cdb3.vmc.netapp.com
System Identifier (SID): cdb3
Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/cdb3/cdb3.log"
for further details.

1. Repeat the same procedures from step 2 to create a container database cdb4 in a separate
   ORACLE_HOME /u01/app/oracle/product/19.0.0/cdb4 with a single PDB.
2. As Oracle user, validate Oracle Restart HA services after DB creation that all databases (cdb3, cdb4)
   are registered with HA services.

   /u01/app/oracle/product/19.0.0/grid/crsctl stat res -t

   output:

   [oracle@ora_02 bin]$ ./crsctl stat res -t
   -------------------------------------------------------------------------
   Name   Target   State   Server    State
   -------------------------------------------------------------------------

### Local Resources

<table>
<thead>
<tr>
<th>Resource</th>
<th>Status</th>
<th>Hostname</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora.DATA.dg</td>
<td>ONLINE</td>
<td>ora_02</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.LISTENER.lsnr</td>
<td>ONLINE</td>
<td>ora_02</td>
<td>Not All Endpoints Re gistered,STABLE</td>
</tr>
<tr>
<td>ora.LOGS.dg</td>
<td>ONLINE</td>
<td>ora_02</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.asm</td>
<td>ONLINE</td>
<td>ora_02</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.ons</td>
<td>OFFLINE</td>
<td>ora_02</td>
<td>STABLE</td>
</tr>
</tbody>
</table>

### Cluster Resources

<table>
<thead>
<tr>
<th>Resource</th>
<th>Status</th>
<th>Hostname</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora.cdb3.db</td>
<td>1</td>
<td>ONLINE</td>
<td>HOME=/u01/app/oracle/product/19.0.0</td>
</tr>
<tr>
<td>/cdb3,STABLE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ora.cdb4.db</td>
<td>1</td>
<td>ONLINE</td>
<td>HOME=/u01/app/oracle/product/19.0.0</td>
</tr>
<tr>
<td>/cdb4,STABLE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ora.cssd</td>
<td>1</td>
<td>ONLINE</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.diskmon</td>
<td>1</td>
<td>OFFLINE</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.driver.afd</td>
<td>1</td>
<td>ONLINE</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.evmd</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. Set the Oracle user `.bash_profile.`

```
vi ~/.bash_profile
```

Add following entries:

```
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db3
export ORACLE_SID=db3
export PATH=$PATH:$ORACLE_HOME/bin
alias asm='export ORACLE_HOME=/u01/app/oracle/product/19.0.0/grid;export ORACLE_SID=+ASM;export PATH=$PATH:$ORACLE_HOME/bin'
alias cdb3='export ORACLE_HOME=/u01/app/oracle/product/19.0.0/cdb3;export ORACLE_SID=cdb3;export PATH=$PATH:$ORACLE_HOME/bin'
alias cdb4='export ORACLE_HOME=/u01/app/oracle/product/19.0.0/cdb4;export ORACLE_SID=cdb4;export PATH=$PATH:$ORACLE_HOME/bin'
```

4. Validate the CDB/PDB created for cdb3.

```
cdb3
```

```
[oracle@ora_02 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Mon Oct 9 08:19:20 2023
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.18.0.0.0

SQL> select name, open_mode from v$database;
```
NAME      OPEN_MODE
--------- --------------------
CDB3      READ WRITE

SQL> show pdbs

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>CON_NAME</th>
<th>OPEN_MODE</th>
<th>RESTRICTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>READ ONLY</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>CDB3_PDB1</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>4</td>
<td>CDB3_PDB2</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>5</td>
<td>CDB3_PDB3</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
</tbody>
</table>

SQL>

SQL> select name from v$datafile;

NAME
-----------------------------------------------
+DATA/CDB3/DATAFILE/system.257.1149420273
+DATA/CDB3/DATAFILE/sysaux.258.1149420317
+DATA/CDB3/DATAFILE/undotbs1.259.1149420343
+DATA/CDB3/86B637B62FE07A65E053F706E80A27CA/DATAFILE/system.266.1149421085
+DATA/CDB3/86B637B62FE07A65E053F706E80A27CA/DATAFILE/sysaux.267.1149421085
+DATA/CDB3/86B637B62FE07A65E053F706E80A27CA/DATAFILE/undotbs1.268.1149421085
+DATA/CDB3/06FB206DF15ADEE8E065025056B66295/DATAFILE/system.272.1149422017
+DATA/CDB3/06FB206DF15ADEE8E065025056B66295/DATAFILE/sysaux.273.1149422017
+DATA/CDB3/06FB206DF15ADEE8E065025056B66295/DATAFILE/undotbs1.271.1149422017
+DATA/CDB3/06FB206DF15ADEE8E065025056B66295/DATAFILE/users.275.1149422033

NAME
-----------------------------------------------
+DATA/CDB3/06FB21766256DF9AE065025056B66295/DATAFILE/system.277.1149422033
+DATA/CDB3/06FB21766256DF9AE065025056B66295/DATAFILE/sysaux.278.1149422033
+DATA/CDB3/06FB21766256DF9AE065025056B66295/DATAFILE/undotbs1.276.1149422033
5. Validate the CDB/PDB created for cdb4.

```
[cdb4]

[oracle@ora_02 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Mon Oct 9 08:20:26 2023
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.18.0.0.0

SQL> select name, open_mode from v$database;

NAME      OPEN_MODE
--------- -------------------
CDB4      READ WRITE

SQL> show pdbs

CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
2 PDB$SEED                       READ ONLY  NO
```
3 CDB4_PDB READ WRITE NO

SQL>

SQL> select name from v$datafile;

NAME
-----------------------------------------------------------------------------------------------------------------------
+DATA/CDB4/DATAFILE/system.286.1149424943
+DATA/CDB4/DATAFILE/sysaux.287.1149424989
+DATA/CDB4/DATAFILE/undotbs1.288.1149425015
+DATA/CDB4/86B637B62FE07A65E053F706E80A27CA/DATAFILE/system.295.1149425765
+DATA/CDB4/86B637B62FE07A65E053F706E80A27CA/DATAFILE/sysaux.296.1149425765
+DATA/CDB4/DATAFILE/users.289.1149425015
+DATA/CDB4/86B637B62FE07A65E053F706E80A27CA/DATAFILE/undotbs1.297.1149425765
+DATA/CDB4/06FC3070D5E12C23E065025056B66295/DATAFILE/system.301.1149426581
+DATA/CDB4/06FC3070D5E12C23E065025056B66295/DATAFILE/sysaux.302.1149426581
+DATA/CDB4/06FC3070D5E12C23E065025056B66295/DATAFILE/undotbs1.300.1149426581
+DATA/CDB4/06FC3070D5E12C23E065025056B66295/DATAFILE/users.304.1149426597

11 rows selected.

6. Login to each cdb as sysdba with sqlplus and set the DB recovery destination size to the +LOGS disk group size for both cdbs.

   alter system set db_recovery_file_dest_size = 40G scope=both;

7. Login to each cdb as sysdba with sqlplus and enable archive log mode with following command sets in sequence.

   sqlplus /as sysdba

   shutdown immediate;

   startup mount;
This completes Oracle 19c version 19.18 Restart deployment on an Amazon FSx for ONTAP storage and a VMC DB VM. If desired, NetApp recommends relocating the Oracle control file and online log files to the +LOGS disk group.

Oracle backup, restore, and clone with SnapCenter

SnapCenter Setup
SnapCenter relies on a host-side plug-in on database VM to perform application-aware data protection management activities. For detailed information on NetApp SnapCenter plugin for Oracle, refer to this documentation What can you do with the Plug-in for Oracle Database. The following provides high level steps to setup SnapCenter for Oracle database backup, recovery, and clone.

1. Download the latest version of SnapCenter software from NetApp support site: NetApp Support Downloads.

2. As administrator, install latest java JDK from Get Java for desktop applications on SnapCenter server Windows host.

   If Windows server is deployed in a domain environment, add a domain user to SnapCenter server local administrators group and run SnapCenter installation with the domain user.

3. Login to SnapCenter UI via HTTPS port 8846 as installation user to configure SnapCenter for Oracle.

4. Update Hypervisor Settings in global settings.

5. Create Oracle database backup policies. Ideally, create a separate archive log backup policy to allow more frequent backup interval to minimize data loss in the event of a failure.

6. Add database server Credential for SnapCenter access to DB VM. The credential should have sudo privilege on a Linux VM or administrator privilege on a Windows VM.
7. Add FSx ONTAP storage cluster to Storage Systems with cluster management IP and authenticated via fsxadmin user ID.

8. Add Oracle database VM in VMC to Hosts with server credential created in previous step 6.

Ensure that the SnapCenter server name can be resolved to the IP address from the DB VM and DB VM name can be resolved to the IP address from the SnapCenter server.

Database backup
SnapCenter leverages FSx ONTAP volume snapshot for much quicker database backup, restore, or clone compared with traditional RMAN based methodology. The snapshots are application-consistent as the database is put in Oracle backup mode before a snapshot.

1. From the Resources tab, any databases on the VM are auto-discovered after the VM is added to SnapCenter. Initially, the database status shows as Not protected.

2. Create a resources group to backup the database in a logical grouping such as by DB VM etc. In this example, we created an ora_02_data group to do a full online database backup for all databases on VM ora_02. Resources group ora_02_log performs the backup of archived logs only on the VM. Creating a resources group also defines a schedule to execute the backup.

3. Resources group backup can also be triggered manually by clicking on Back up Now and executing the backup with the policy defined in the resources group.
4. The backup job can be monitored at the Monitor tab by clicking on the running job.
5. After a successful backup, the database status shows the job status and the most recent backup time.

6. Click on database to review the backup sets for each database.
Database recovery
SnapCenter provides a number of restore and recovery options for Oracle databases from snapshot backup. In this example, we demonstrate a point in time restoration to recover a dropped table by mistake. On VM ora_02, two databases cdb3, cdb4 share the same +DATA and +LOGS disk groups. Database restoration for one database does not impact the availability of the other database.

1. First, create a test table and insert a row into table to validate a point in time recovery.

```sql
[oracle@ora_02 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Oct 6 14:15:21 2023
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0

SQL> select name, open_mode from v$database;

NAME      OPEN_MODE
--------- --------------------
CDB3      READ WRITE

SQL> show pdbs

CON_ID CON_NAME                       OPEN_MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
2 PDB$SEED                       READ ONLY  NO
3 CDB3_PDB1                      READ WRITE NO
4 CDB3_PDB2                      READ WRITE NO
5 CDB3_PDB3                      READ WRITE NO

SQL>

SQL> alter session set container=cdb3_pdb1;
Session altered.

SQL> create table test (id integer, dt timestamp, event varchar(100));
Table created.
```
SQL> insert into test values(1, sysdate, 'test oracle recovery on guest mounted fsx storage to VMC guest vm ora_02');

1 row created.

SQL> commit;

Commit complete.

SQL> select * from test;

<table>
<thead>
<tr>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>DT</td>
</tr>
<tr>
<td>06-OCT-23 03.18.24.000000 PM</td>
</tr>
<tr>
<td>EVENT</td>
</tr>
<tr>
<td>test oracle recovery on guest mounted fsx storage to VMC guest vm ora_02</td>
</tr>
</tbody>
</table>

SQL> select current_timestamp from dual;

<table>
<thead>
<tr>
<th>CURRENT_TIMESTAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>06-OCT-23 03.18.53.996678 PM -07:00</td>
</tr>
</tbody>
</table>

2. We run a manual snapshot backup from SnapCenter. Then drop the table.
SQL> drop table test;
Table dropped.
SQL> commit;
Commit complete.
SQL> select current_timestamp from dual;
CURRENT_TIMESTAMP
-----------------------------------
----------
06-OCT-23 03.26.30.169456 PM -07:00
SQL> select * from test;
select * from test
*
ERROR at line 1:
ORA-00942: table or view does not exist

3. From backup set created from last step, take a note of the SCN number of log backup. Click on Restore to launch restore-recover workflow.

4. Choose restore scope.
5. Choose recovery scope up to the log SCN from last full database backup.
6. Specify any optional pre-scripts to run.
7. Specify any optional after-script to run.
8. Send a job report if desired.
9. Review the summary and click on Finish to launch the restoration and recovery.
10. From Oracle Restart grid control, we observe that while cdb3 is under restoration and recovery cdb4 is online and available.
11. From Monitor tab, open the job to review the details.
12. From DB VM ora_02, validate the dropped table is recovered after a successful recovery.

[oracle@ora_02 bin]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Oct 6 17:01:28 2023
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.18.0.0.0

SQL> select name, open_mode from v$database;

NAME      OPEN_MODE
--------- ---------------------
CDB3      READ WRITE

SQL> show pdbs

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>CON_NAME</th>
<th>OPEN_MODE</th>
<th>RESTRICTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>READ ONLY</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>CDB3_PDB1</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>4</td>
<td>CDB3_PDB2</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>5</td>
<td>CDB3_PDB3</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
</tbody>
</table>

SQL> alter session set container=CDB3_PDB1;

Session altered.

SQL> select * from test;

<table>
<thead>
<tr>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EVENT</th>
<th>------------</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>06-OCT-23 03.18.24.000000 PM test oracle recovery on guest mounted fsx storage to VMC guest vm ora_02</td>
</tr>
</tbody>
</table>

SQL> select current_timestamp from dual;

CURRENT_TIMESTAMP

| 06-OCT-23 05.02.20.382702 PM -07:00 |

SQL>
Database clone
In this example, the same backup sets is used to clone a database on the same VM in a different ORACLE_HOME. The procedures are equally applicable to clone a database from the backup to separate VM in VMC if needed.

1. Open the database cdb3 backup list. From a data backup of choice, click on Clone button to launch database clone workflow.

2. Name the clone database SID.
3. Select a VM in VMC as the target database host. Identical Oracle version should have been installed and configured on the host.
4. Select the proper ORACLE_HOME, user and group on the target host. Keep credential at default.
5. Change clone database parameters to meet configuration or resources requirements for the clone database.
6. Choose recovery scope. Until Cancel recovers the clone up to last available log file in the backup set.
7. Review the summary and launch the clone job.
8. Monitor the clone job execution from Monitor tab.
9. Cloned database is immediately registered in SnapCenter.
10. From DB VM ora_02, the cloned database is also registered in Oracle Restart grid control and the dropped test table is recovered in the cloned database cdb3tst as shown below.

```
[oracle@ora_02 ~]$ /u01/app/oracle/product/19.0.0/grid/bin/crsctl
stat res -t

Name        Target  State          Server   State
details

Local Resources

ora.DATA.dg  ONLINE  ONLINE       ora_02   STABLE
ora.LISTENER.1snr ONLINE  INTERMEDIATE ora_02  Not All Endpoints Re
gistered,STABLE
ora.LOGS.dg   ONLINE  ONLINE       ora_02   STABLE
ora.SC_2090922_CDB3TST.dg ONLINE  ONLINE       ora_02   STABLE
ora.asm      ONLINE  ONLINE       ora_02   STABLE
ora.ons      OFFLINE OFFLINE      ora_02  STABLE

Cluster Resources

ora.cdb3.db   1 ONLINE  ONLINE       ora_02
Open,HOME=/u01/app/oracle/product/19.0.0

/oracle/product/19.0.0
ora.cdb3tst.db 1 ONLINE  ONLINE       ora_02
Open,HOME=/u01/app/oracle/product/19.0.0
```
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.18.0.0.0

SQL> select name, open_mode from v$database;

NAME      OPEN_MODE
--------- -------------------
CDB3TST   READ WRITE

SQL> show pdbs

CON_ID CON_NAME OPEN_MODE RESTRICTED
---------- ---------- ---------------
This completes the demonstration of SnapCenter backup, restore, and clone of Oracle database in VMC SDDC on AWS.

Where to find additional information

To learn more about the information described in this document, review the following documents and/or websites:

- VMware Cloud on AWS Documentation
- Installing Oracle Grid Infrastructure for a Standalone Server with a New Database Installation
- Installing and Configuring Oracle Database Using Response Files
  https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/installing-and-configuring-oracle-
Purpose

Oracle Data Guard ensures high availability, data protection, and disaster recovery for enterprise data in a primary database and standby database replication configuration. Oracle Active Data Guard empowers users to access standby databases while data replication is active from the primary database to standby databases. Data Guard is a feature of Oracle Database Enterprise Edition. It does not require separate licensing. On the other hand, Active Data Guard is an Oracle Database Enterprise Edition Option therefore requires separate licensing. Multiple standby databases can receive data replication from a primary database in the Active Data Guard setup. However, each additional standby database requires an Active Data Guard license and extra storage as the size of primary database. The operational costs add up quickly.

If you are keen on cutting back cost of your Oracle database operation and are planning to set up an Active Data Guard in AWS, you should consider an alternative. Instead of Active Data Guard, use Data Guard to replicate from primary database to a single physical standby database on Amazon FSx ONTAP storage. Subsequently, multiple copies of this standby database can be cloned and opened for read/write access to serve many other use cases such as reporting, development, test etc. The net results effectively deliver functionalities of Active Data Guard while eliminating Active Data Guard license and extra storage cost for each additional standby database. In this documentation, we demonstrate how to setup an Oracle Data Guard with your existing primary database in AWS and place physical standby database on Amazon FSx ONTAP storage. The standby database is backed up via snapshot and cloned for read/write access for use cases as desired.

This solution addresses the following use cases:

- Oracle Data Guard between a primary database on any storage in AWS to standby database on Amazon FSx ONTAP storage.
- Clone the standby database while closed for data replication to serve use cases such as reporting, dev, test, etc.

Audience

This solution is intended for the following people:

- A DBA who set up Oracle Active Data Guard in AWS for high availability, data protection, and disaster recovery.
- A database solution architect interested in Oracle Active Data Guard configuration in the AWS cloud.
- A storage administrator who manages AWS FSx ONTAP storage that supports Oracle Data Guard.
- An application owner who like to stand up Oracle Data Guard in AWS FSx/EC2 environment.

Solution test and validation environment

The testing and validation of this solution was performed in an AWS FSx ONTAP and EC2 lab environment that...
Architecture

Oracle Data Guard Deployment with Amazon FSx for ONTAP

Hardware and software components

**Hardware**

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSx ONTAP storage</td>
<td>Current version offered by AWS</td>
<td>One FSx HA cluster in the same VPC and availability zone</td>
</tr>
<tr>
<td>EC2 instance for compute</td>
<td>t2.xlarge/4vCPU/16G</td>
<td>Three EC2 T2 xlarge EC2 instances, one as primary DB server, one as standby DB server, and the third as a clone DB server</td>
</tr>
</tbody>
</table>

**Software**

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>RedHat Linux</td>
<td>RHEL-8.6.0_HVM-20220503-x86_64-2-Hourly2-GP2</td>
<td>Deployed RedHat subscription for testing</td>
</tr>
<tr>
<td>Oracle Grid Infrastructure</td>
<td>Version 19.18</td>
<td>Applied RU patch p34762026_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle Database</td>
<td>Version 19.18</td>
<td>Applied RU patch p34765931_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle OPatch</td>
<td>Version 12.2.0.1.36</td>
<td>Latest patch p6880880_190000_Linux-x86-64.zip</td>
</tr>
</tbody>
</table>
Oracle Data Guard configuration with hypothetical NY to LA DR setup

<table>
<thead>
<tr>
<th>Database</th>
<th>DB_UNIQUE_NAME</th>
<th>Oracle Net Service Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>db1_NY</td>
<td>db1_NY.demo.netapp.com</td>
</tr>
<tr>
<td>Physical Standby</td>
<td>db1_LA</td>
<td>db1_LA.demo.netapp.com</td>
</tr>
</tbody>
</table>

Key factors for deployment consideration

- **How Oracle Standby Database FlexClone Works.** AWS FSx ONTAP FlexClone provides shared copies of the same standby database volumes that are writable. The copies of the volumes are actually pointers that link back to original data blocks until a new write initiates on the clone. ONTAP then allocates new storage blocks for the new writes. Any read IOs are serviced by original data blocks under active replication. Thus, the clone are very storage efficient that can be used for many other use cases with minimal and incremental new storage allocation for new write IOs. This provides tremendous storage cost saving by substantially reducing Active Data Guard storage footprint. NetApp recommends to minimize FlexClone activities in the event of database switching over from primary storage to standby FSx storage in order to maintain Oracle performance at high level.

- **Oracle Software Requirements.** In general, a physical standby database must have the same Database Home version as the primary database including Patch Set Exceptions (PSEs), Critical Patch Updates (CPUs), and Patch Set Updates (PSUs), unless an Oracle Data Guard Standby-First Patch Apply process is in progress (as described in My Oracle Support note 1265700.1 at support.oracle.com)

- **Standby Database Directory Structure Considerations.** If possible, the data files, log files, and control files on the primary and standby systems should have the same names and path names and use Optimal Flexible Architecture (OFA) naming conventions. The archival directories on the standby database should also be identical between sites, including size and structure. This strategy allows other operations such as backups, switchovers, and failovers to execute the same set of steps, reducing the maintenance complexity.

- **Force Logging Mode.** To protect against unlogged direct writes in the primary database that cannot be propagated to the standby database, turn on FORCE LOGGING at the primary database before performing data file backups for standby creation.

- **Database Storage Management.** For operational simplicity, Oracle recommends that when you set up Oracle Automatic Storage Management (Oracle ASM) and Oracle Managed Files (OMF) in an Oracle Data Guard configuration that you set it up symmetrically on the primary and standby database(s).

- **EC2 compute instances.** In these tests and validations, we used an AWS EC2 t2.xlarge instance as the Oracle database compute instance. NetApp recommends using a M5 type EC2 instance as the compute instance for Oracle in production deployment because it is optimized for database workload. You need to size the EC2 instance appropriately for the number of vCPUs and the amount of RAM based on actual workload requirements.

- **FSx storage HA clusters single- or multi-zone deployment.** In these tests and validations, we deployed an FSx HA cluster in a single AWS availability zone. For production deployment, NetApp recommends deploying an FSx HA pair in two different availability zones. An FSx cluster is alway provisioned in a HA pair that is sync mirrored in a pair of active-passive file systems to provide storage-level redundancy. Multi-zone deployment further enhances high availability in the event of failure in a single AWS zone.

- **FSx storage cluster sizing.** An Amazon FSx for ONTAP storage file system provides up to 160,000 raw SSD IOPS, up to 4GBps throughput, and a maximum of 192TiB capacity. However, you can size the cluster in terms of provisioned IOPS, throughput, and the storage limit (minimum 1,024 GiB) based on your actually requirements at the time of deployment. The capacity can be adjusted dynamically on the fly without affecting application availability.
Solution deployment

It is assumed that you already have your primary Oracle database deployed in AWS EC2 environment within a VPC as the starting point for setting up Data Guard. The primary database is deployed using Oracle ASM for storage management. Two ASM disk groups - +DATA and +LOGS are created for Oracle data files, log files, and control file etc. For details on Oracle deployment in AWS with ASM, please refer to following technical reports for help.

- Oracle Database Deployment on EC2 and FSx Best Practices
- Oracle Database Deployment and Protection in AWS FSx/EC2 with iSCSI/ASM
- Oracle 19c in Standalone Restart on AWS FSx/EC2 with NFS/ASM

Your primary Oracle database can be running either on an FSx ONTAP or any other storage of choices within the AWS EC2 ecosystem. The following section provides step-by-step deployment procedures for setting up Oracle Data Guard between a primary EC2 DB instance with ASM storage to a standby EC2 DB instance with ASM storage.

Prerequisites for deployment

Deployment requires the following prerequisites.

1. An AWS account has been set up, and the necessary VPC and network segments have been created within your AWS account.

2. From the AWS EC2 console, you need to deploy minimum three EC2 Linux instances, one as the primary Oracle DB instance, one as standby Oracle DB instance, and an clone target DB instance for reporting, dev, and test etc. See the architecture diagram in the previous section for more details about the environment setup. Also review the AWS User Guide for Linux instances for more information.

3. From the AWS EC2 console, deploy Amazon FSx for ONTAP storage HA clusters to host Oracle volumes that stores the Oracle standby database. If you are not familiar with the deployment of FSx storage, see the documentation Creating FSx for ONTAP file systems for step-by-step instructions.

4. Steps 2 and 3 can be performed using the following Terraform automation toolkit, which creates an EC2 instance named ora_01 and an FSx file system named fsx_01. Review the instruction carefully and change the variables to suit your environment before execution. The template can be easily revised for your own deployment requirements.

```
git clone https://github.com/NetApp-Automation/na_aws_fsx_ec2_deploy.git
```

Ensure that you have allocated at least 50G in EC2 instance root volume in order to have sufficient space to stage Oracle installation files.

Prepare the primary database for Data Guard
In this demonstration, we have setup a primary Oracle database called db1 on the primary EC2 DB instance with two ASM disk groups in standalone Restart configuration with data files in ASM disk group +DATA and flash recovery area in ASM disk group +LOGS. Following illustrates the detailed procedures for setting up primary database for Data Guard. All steps should be executed as database owner - oracle user.

1. Primary database db1 configuration on primary EC2 DB instance ip-172-30-15-45. The ASM disk groups can be on any type of storage within EC2 ecosystem.

```
[oracle@ip-172-30-15-45 ~]$ cat /etc/oratab

# This file is used by ORACLE utilities. It is created by root.sh
# and updated by either Database Configuration Assistant while creating
# a database or ASM Configuration Assistant while creating ASM instance.

# A colon, ':', is used as the field terminator. A new line terminates
# the entry. Lines beginning with a pound sign, '#', are comments.
#
# Entries are of the form:
#   $ORACLE_SID:$ORACLE_HOME:<N|Y>:
#
# The first and second fields are the system identifier and home
directory of the database respectively. The third field indicates
# to the dbstart utility that the database should, "Y", or should not,
# "N", be brought up at system boot time.
#
# Multiple entries with the same $ORACLE_SID are not allowed.
#
+ASM:/u01/app/oracle/product/19.0.0/grid:N
db1:/u01/app/oracle/product/19.0.0/db1:N
```

```
[oracle@ip-172-30-15-45 ~]$ /u01/app/oracle/product/19.0.0/grid/bin/crsctl stat res -t

<table>
<thead>
<tr>
<th>Name</th>
<th>Target</th>
<th>State</th>
<th>Server</th>
<th>State details</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local Resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

874
<table>
<thead>
<tr>
<th>Service</th>
<th>State</th>
<th>IP Address</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora.DATA.dg</td>
<td>ONLINE</td>
<td>ip-172-30-15-45</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.LISTENER.lsnr</td>
<td>ONLINE</td>
<td>ip-172-30-15-45</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.LOGS.dg</td>
<td>ONLINE</td>
<td>ip-172-30-15-45</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.asm</td>
<td>ONLINE</td>
<td>ip-172-30-15-45</td>
<td>STABLE</td>
</tr>
<tr>
<td>Started,STABLE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ora.ons</td>
<td>OFFLINE</td>
<td>ip-172-30-15-45</td>
<td>STABLE</td>
</tr>
</tbody>
</table>

Cluster Resources

<table>
<thead>
<tr>
<th>Service</th>
<th>State</th>
<th>IP Address</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora.cssd</td>
<td>1</td>
<td>ONLINE  ip-172-30-15-45</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.db1.db</td>
<td>1</td>
<td>ONLINE ip-172-30-15-45</td>
<td>STABLE</td>
</tr>
<tr>
<td>Open,HOME=/u01/app/oracle/product/19.0.0/db1,STABLE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ora.diskmon</td>
<td>1</td>
<td>OFFLINE ip-172-30-15-45</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.driver.afd</td>
<td>1</td>
<td>ONLINE ip-172-30-15-45</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.evmd</td>
<td>1</td>
<td>ONLINE ip-172-30-15-45</td>
<td>STABLE</td>
</tr>
</tbody>
</table>

2. From sqlplus, enable forced logging on primary.

`alter database force logging;`

3. From sqlplus, enable flashback on primary. Flashback allows easy reinstate primary database as a standby after a failover.

`alter database flashback on;`
4. Configure redo transport authentication using Oracle password file - create a pwd file on the primary using orapwd utility if not set and copy over to standby database $ORACLE_HOME/dbs directory.

5. Create standby redo logs on the primary DB with same size as current online log file. Log groups are one more than online log file groups. The primary database can then quickly transition to the standby role and begin receiving redo data, if necessary.

```sql
alter database add standby logfile thread 1 size 200M;
```

Validate after standby logs addition:

```sql
SQL> select group#, type, member from v$logfile;
```

<table>
<thead>
<tr>
<th>GROUP#</th>
<th>TYPE</th>
<th>MEMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ONLINE</td>
<td>+DATA/DB1/ONLINELOG/group_3.264.1145821513</td>
</tr>
<tr>
<td>2</td>
<td>ONLINE</td>
<td>+DATA/DB1/ONLINELOG/group_2.263.1145821513</td>
</tr>
<tr>
<td>1</td>
<td>ONLINE</td>
<td>+DATA/DB1/ONLINELOG/group_1.262.1145821513</td>
</tr>
<tr>
<td>4</td>
<td>STANDBY</td>
<td>+DATA/DB1/ONLINELOG/group_4.286.1146082751</td>
</tr>
<tr>
<td>4</td>
<td>STANDBY</td>
<td>+LOGS/DB1/ONLINELOG/group_4.258.1146082753</td>
</tr>
<tr>
<td>5</td>
<td>STANDBY</td>
<td>+DATA/DB1/ONLINELOG/group_5.287.1146082819</td>
</tr>
<tr>
<td>5</td>
<td>STANDBY</td>
<td>+LOGS/DB1/ONLINELOG/group_5.260.1146082821</td>
</tr>
<tr>
<td>6</td>
<td>STANDBY</td>
<td>+DATA/DB1/ONLINELOG/group_6.288.1146082825</td>
</tr>
<tr>
<td>6</td>
<td>STANDBY</td>
<td>+LOGS/DB1/ONLINELOG/group_6.261.1146082827</td>
</tr>
<tr>
<td>7</td>
<td>STANDBY</td>
<td>+DATA/DB1/ONLINELOG/group_7.289.1146082835</td>
</tr>
<tr>
<td>7</td>
<td>STANDBY</td>
<td>+LOGS/DB1/ONLINELOG/group_7.262.1146082835</td>
</tr>
</tbody>
</table>

11 rows selected.

6. From sqlplus, create a pfile from spfile for editing.

```sql
create pfile='/home/oracle/initdb1.ora' from spfile;
```

7. Revise the pfile and add following parameters.
8. From sqlplus, create spfile in ASM +DATA directory from revised pfile in /home/oracle directory.

```sql
create spfile='+DATA' from pfile='/home/oracle/initdb1.ora';
```

9. Locate the newly created spfile under +DATA disk group(using asmcmd utility if necessary). Use srvctl to modify grid to start database from new spfile as shown below.
10. Modify tnsnames.ora to add db_unique_name for name resolution.
# tnsnames.ora Network Configuration File:
/u01/app/oracle/product/19.0.0/db1/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

db1_NY =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = ip-172-30-15-45.ec2.internal)(PORT = 1521))
    (CONNECT_DATA =
     (SERVER = DEDICATED)
     (SID = db1)
    )
  )

db1_LA =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = ip-172-30-15-67.ec2.internal)(PORT = 1521))
    (CONNECT_DATA =
     (SERVER = DEDICATED)
     (SID = db1)
    )
  )

LISTENER_DB1 =
  (ADDRESS = (PROTOCOL = TCP)(HOST = ip-172-30-15-45.ec2.internal)(PORT = 1521))

11. Add data guard service name db1_NY_DGMGRL.demo.netapp for primary database to listener.ora file.
LISTENER =
  (DESCRIPTION_LIST =
   (DESCRIPTION =
     (ADDRESS = (PROTOCOL = TCP)(HOST = ip-172-30-15-45.ec2.internal)(PORT = 1521))
     (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
   )
  )

SID_LIST_LISTENER =
  (SID_LIST =
   (SID_DESC =
     (GLOBAL_DBNAME = db1_NY_DGMGRL.demo.netapp.com)
     (ORACLE_HOME = /u01/app/oracle/product/19.0.0/db1)
     (SID_NAME = db1)
   )
  )

ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER=ON              # line added by Agent
VALID_NODE_CHECKING_REGISTRATION_LISTENER=ON            # line added by Agent

1. Shutdown and restart database with srvctl and validate that data guard parameters are now active.

   srvctl stop database -d db1

   srvctl start database -d db1

This completes primary database setup for Data Guard.

Prepare standby database and activate Data Guard
Oracle Data Guard requires OS kernel configuration and Oracle software stacks including patch sets on standby EC2 DB instance to match with primary EC2 DB instance. For easy management and simplicity, the standby EC2 DB instance database storage configuration ideally should match with the primary EC2 DB instance as well, such as the name, number and size of ASM disk groups. Following are detail procedures for setting up the standby EC2 DB instance for Data Guard. All commands should be executed as oracle owner user id.

1. First, review the configuration of the primary database on primary EC2 instance. In this demonstration, we have setup a primary Oracle database called db1 on the primary EC2 DB instance with two ASM disk groups +DATA and +LOGS in standalone Restart configuration. The primary ASM disk groups may be on any type of storage within EC2 ecosystem.

2. Follow procedures in documentation TR-4965: Oracle Database Deployment and Protection in AWS FSx/EC2 with iSCSI/ASM to install and configure grid and Oracle on standby EC2 DB instance to match with primary database. The database storage should be provisioned and allocated to standby EC2 DB instance from FSx ONTAP with same storage capacity as primary EC2 DB instance.

   ![Info]

Stop at step 10 in Oracle database installation section. The standby database will be instantiated from primary database using dbca database duplication function.

3. Once Oracle software is installed and configured, from standby $ORACLE_HOME dbs directory, copy oracle password from primary database.

   ```
   scp oracle@172.30.15.45:/u01/app/oracle/product/19.0.0/db1/dbs/orapwdb1 .
   ```

4. Create tnsnames.ora file with following entries.
5. Add DB data guard service name to listener.ora file.
# Backup file is /u01/app/oracle/crsdata/ip-172-30-15-67/output/listener.ora.bak.ip-172-30-15-67.oracle line added by Agent

# listener.ora Network Configuration File:
/u01/app/oracle/product/19.0.0/grid/network/admin/listener.ora

# Generated by Oracle configuration tools.

LISTENER =
   (DESCRIPTION_LIST =
      (DESCRIPTION =
         (ADDRESS = (PROTOCOL = TCP)(HOST = ip-172-30-15-67.ec2.internal)(PORT = 1521))
         (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
      )
   )

SID_LIST_LISTENER =
   (SID_LIST =
      (SID_DESC =
         (GLOBAL_DBNAME = db1_LA_DGMGRL.demo.netapp.com)
         (ORACLE_HOME = /u01/app/oracle/product/19.0.0/db1)
         (SID_NAME = db1)
      )
   )

ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER=ON  # line added by Agent
VALID_NODE_CHECKING_REGISTRATION_LISTENER=ON  # line added by Agent

6. Set oracle home and path.

```bash
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db1
```

```bash
export PATH=$PATH:$ORACLE_HOME/bin
```

7. Use dbca to instantiate standby database from primary database db1.
SQL> show parameter name

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdb_cluster_name</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>cell_offloadgroup_name</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>db_file_name_convert</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>db_name</td>
<td>string</td>
<td>db1</td>
</tr>
<tr>
<td>db_unique_name</td>
<td>string</td>
<td>db1_LA</td>
</tr>
<tr>
<td>global_names</td>
<td>boolean</td>
<td>FALSE</td>
</tr>
<tr>
<td>instance_name</td>
<td>string</td>
<td>db1</td>
</tr>
<tr>
<td>lock_name_space</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>log_file_name_convert</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>pdb_file_name_convert</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>processor_group_name</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>service_names</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>db1_LA.demo.netapp.com</td>
<td>string</td>
<td></td>
</tr>
</tbody>
</table>

SQL>

SQL> show parameter log_archive_config

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>log_archive_config</td>
<td>string</td>
<td>DG_CONFIG=(db1_NY,db1_LA)</td>
</tr>
</tbody>
</table>

SQL> show parameter fal_server

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>fal_server</td>
<td>string</td>
<td>db1_NY</td>
</tr>
</tbody>
</table>

SQL> select name from v$datafile;

<table>
<thead>
<tr>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>+DATA/DB1_LA/DATAFILE/system.261.1146248215</td>
</tr>
<tr>
<td>+DATA/DB1_LA/DATAFILE/sysaux.262.1146248231</td>
</tr>
<tr>
<td>+DATA/DB1_LA/DATAFILE/undotbs1.263.1146248247</td>
</tr>
<tr>
<td>+DATA/DB1_LA/03C5C01A66EE9797E0632D0F1EAC5F59/DATAFILE/system.264.11</td>
</tr>
</tbody>
</table>
SQL> select name from v$controlfile;

NAME
------------------------------------------
+DATA/DB1_LA/CONTROLFILE/current.260.11
+LOGS/DB1_LA/CONTROLFILE/current.257.11

SQL> select name from v$tempfile;

NAME
------------------------------------------
+DATA/DB1_LA/CONTROLFILE/current.260.11
+LOGS/DB1_LA/CONTROLFILE/current.257.11

19 rows selected.
9. Restart standby database in mount stage and execute following command to activate standby database managed recovery.

```sql
SQL> select group#, type, member from v$logfile order by 2, 1;

GROUP#  TYPE    MEMBER
---------- ------- ----------------------------
        ONLINE  +LOGS/DB1_LA/ONLINELOG/group_1.259.1146248349
        ONLINE  +DATA/DB1_LA/ONLINELOG/group_1.280.1146248347
        ONLINE  +DATA/DB1_LA/ONLINELOG/group_2.281.1146248351
        ONLINE  +LOGS/DB1_LA/ONLINELOG/group_2.258.1146248353
        ONLINE  +DATA/DB1_LA/ONLINELOG/group_3.282.1146248355
        ONLINE  +LOGS/DB1_LA/ONLINELOG/group_3.260.1146248355
        STANDBY +DATA/DB1_LA/ONLINELOG/group_4.283.1146248357
        STANDBY +LOGS/DB1_LA/ONLINELOG/group_4.261.1146248359
        STANDBY +LOGS/DB1_LA/ONLINELOG/group_5.284.1146248361
        STANDBY +LOGS/DB1_LA/ONLINELOG/group_5.262.1146248363
        STANDBY +LOGS/DB1_LA/ONLINELOG/group_6.263.1146248365
        STANDBY +LOGS/DB1_LA/ONLINELOG/group_6.285.1146248365
        STANDBY +LOGS/DB1_LA/ONLINELOG/group_7.264.1146248369
        STANDBY +DATA/DB1_LA/ONLINELOG/group_7.286.1146248367

14 rows selected.
```

```sql
SQL> select name, open_mode from v$database;

NAME      OPEN_MODE
--------- ---------------------
DB1      READ ONLY
```

9. Restart standby database in mount stage and execute following command to activate standby database managed recovery.
alter database recover managed standby database disconnect from session;

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.

Total System Global Area 8053062944 bytes
Fixed Size                  9182496 bytes
Variable Size            1291845632 bytes
Database Buffers         6744440832 bytes
Redo Buffers                7593984 bytes
Database mounted.
SQL> alter database recover managed standby database disconnect from session;
Database altered.

10. Validate the standby database recovery status. Notice the recovery logmerger in APPLYING_LOG action.
This completes the Data Guard protection setup for db1 from primary to standby with managed standby recovery enabled.

**Setup Data Guard Broker**
Oracle Data Guard broker is a distributed management framework that automates and centralizes the creation, maintenance, and monitoring of Oracle Data Guard configurations. Following section demonstrate how to setup Data Guard Broker to manage Data Guard environment.

1. Start data guard broker on both primary and standby databases with following command via sqlplus.

   ```sql
   alter system set dg_broker_start=true scope=both;
   ```

2. From primary database, connect to Data Guard Broker as SYSDBA.

   ```
   [oracle@ip-172-30-15-45 db1]$ dgmgrl sys@db1_NY
   DGMGRL for Linux: Release 19.0.0.0.0 - Production on Wed Aug 30
   19:34:14 2023
   Version 19.18.0.0.0

   Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.

   Welcome to DGMGRL, type "help" for information.
   Password:
   Connected to "db1_NY"
   Connected as SYSDBA.
   ```

3. Create and enable Data Guard Broker configuration.
DGMGRL> create configuration dg_config as primary database is db1_NY
connect identifier is db1_NY;
Configuration "dg_config" created with primary database "db1_NY"
DGMGRL> add database db1_LA as connect identifier is db1_LA;
Database "db1_la" added
DGMGRL> enable configuration;
Enabled.
DGMGRL> show configuration;

Configuration - dg_config

Protection Mode: MaxPerformance
Members:
  db1_ny - Primary database
    db1_la - Physical standby database

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS  (status updated 28 seconds ago)

4. Validate database status within Data Guard Broker management framework.
DGMGRL> show database db1_ny;

Database - db1_ny

Role: PRIMARY
Intended State: TRANSPORT-ON
Instance(s): 
  db1

Database Status:
SUCCESS

DGMGRL> show database db1_la;

Database - db1_la

Role: PHYSICAL STANDBY
Intended State: APPLY-ON
Transport Lag: 0 seconds (computed 1 second ago)
Apply Lag: 0 seconds (computed 1 second ago)
Average Apply Rate: 2.00 KByte/s
Real Time Query: OFF
Instance(s): 
  db1

Database Status:
SUCCESS

DGMGRL>

In the event of a failure, Data Guard Broker can be used to failover primary database to standby instantaniouly.

Clone standby database for other use cases
The key benefit of staging standby database on AWS FSx ONTAP in Data Guard is that it can be FlexCloned to serve many other use cases with minimal additional storage investment. In the following section, we demonstrate how to snapshot and clone the mounted and under recovery standby database volumes on FSx ONTAP for other purposes, such as DEV, TEST, REPORT, etc., using the NetApp SnapCenter tool.

Following are high level procedures to clone a READ/WRITE database from the managed physical standby database in Data Guard using SnapCenter. For detail instructions on how to setup and configure SnapCenter, please refer to Hybrid Cloud Database Solutions with SnapCenter relavant Oracle sections.

1. We begin with creating a test table and inserting a row into the test table on primary database. We will then validate if the transaction traverse down to standby and finally the clone.

   ```sql
   [oracle@ip-172-30-15-45 db1]$ sqlplus / as sysdba
   SQL*Plus: Release 19.0.0.0.0 - Production on Thu Aug 31 16:35:53 2023
   Version 19.18.0.0.0
   Copyright (c) 1982, 2022, Oracle.  All rights reserved.
   Connected to:
   Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
   Version 19.18.0.0.0
   SQL> alter session set container=db1_pdb1;
   Session altered.
   SQL> create table test(
         2  id integer,
         3  dt timestamp,
         4  event varchar(100));
   Table created.
   SQL> insert into test values(1, sysdate, 'a test transaction on primary database db1 and ec2 db host: ip-172-30-15-45.ec2.internal');
   1 row created.
   SQL> commit;
   Commit complete.
   ```
SQL> select * from test;

   ID  
  -------

   DT  
  ----------------------------------------

   EVENT  
  ----------------------------------------

  1
31-AUG-23  04.49.29.000000 PM
a test transaction on primary database db1 and ec2 db host: ip-172-30-15-45.ec2.
internal

SQL> select instance_name, host_name from v$instance;

 INSTANCE_NAME  
 ---------------

 HOST_NAME  
 --------------------------

 db1
ip-172-30-15-45.ec2.internal

2. Add FSx storage cluster to Storage Systems in SnapCenter with FSx cluster management IP and fsxadmin credential.

3. Add AWS ec2-user to Credential in Settings.
4. Add standby EC2 DB instance and clone EC2 DB instance to Hosts.

The clone EC2 DB instance should have similar Oracle software stacks installed and configured. In our test case, the grid infrastructure and Oracle 19C installed and configured but no database created.

5. Create a backup policy that is tailored for offline/mount full database backup.

6. Apply backup policy to protect standby database in Resources tab.

7. Click on database name to open the database backups page. Select a backup to be used for
database clone and click on **Clone** button to launch clone workflow.

8. Select **Complete Database Clone** and name the clone instance SID.

9. Select the clone host, which hosts the cloned database from standby DB. Accept the default for data files, control files, and redo logs. Two ASM disk groups will be created on the clone host that are corresponding to the disk groups on standby database.
10. No database credentials are needed for OS based authentication. Match Oracle home setting with what is configured on the clone EC2 database instance.
11. Change clone database parameters if needed and specify scripts to run before clone if any.
12. Enter SQL to run after clone. In the demo, we executed commands to turn off database archive mode for a dev/test/report database.
13. Configure email notification if desired.
14. Review the summary, click Finish to start the clone.
Monitor clone job in Monitor tab. We observed that it took around 8 minutes to clone a database about 300GB in database volume size.
16. Validate the clone database from SnapCenter, which is immediately registered in Resources tab right after clone operation.

17. Query the clone database from clone EC2 instance. We validated that test transaction that occurred in primary database had traversed down to clone database.
SQL*Plus: Release 19.0.0.0.0 - Production on Wed Sep 6 16:41:41 2023
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0

SQL> select name, open_mode, log_mode from v$database;

NAME      OPEN_MODE            LOG_MODE
--------- -------------------- ------------
DB1DEV    READ WRITE           NOARCHIVELOG

SQL> select instance_name, host_name from v$instance;

INSTANCE_NAME
-------------
HOST_NAME
-------------
db1dev
ip-172-30-15-126.ec2.internal

SQL> alter session set container=db1_pdb1;

Session altered.

SQL> select * from test;

ID
---
DT
-------------
EVENT
-------------

904
This completes the clone and validation of a new Oracle database from standby database in Data Guard on FSx storage for DEV, TEST, REPORT or any other use cases. Multiple Oracle databases can be cloned off the same standby database in Data Guard.

Where to find additional information

To learn more about the information described in this document, review the following documents and/or websites:

- Data Guard Concepts and Administration
  
  https://docs.oracle.com/en/database/oracle/oracle-database/19/sbydb/index.html#Oracle%C2%AE-Data-Guard

- WP-7357: Oracle Database Deployment on EC2 and FSx Best Practices
  

- Amazon FSx for NetApp ONTAP
  
  https://aws.amazon.com/fsx/netapp-ontap/

- Amazon EC2
  
  https://aws.amazon.com/pm/ec2/?trk=36c6da98-7b20-48fa-8225-4784bcd9843&sc_channel=ps&s_kwcid=AL!4422!3!467723097970le!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARlsAJzSrdqwQrghn6l71jiWzSeAT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:G:s&s_kwcid=AL!4422!3!467723097970le!!g!!aws%20ec2

TR-4973: Quick Recovery and Clone of Oracle VLDB with Incremental Merge on AWS FSx ONTAP

Allen Cao, Niyaz Mohamed, NetApp

Purpose

Recovering a Very Large Database (VLDB) in Oracle using the Oracle Recovery Manager (RMAN) backup tool can be a highly challenging task. The database restoration process from backup media in the event of a failure can be time-consuming, delaying the database recovery and potentially impacting your Service Level Agreement (SLA) significantly. However, starting from version 10g, Oracle introduced a RMAN feature that allows users to create staged image copies of the Oracle database data files on additional disk storage located on the DB server host. These image copies can be incrementally updated using RMAN on a daily basis. In the
case of a failure, the Database Administrator (DBA) can swiftly switch the Oracle database from the failed media to the image copy, eliminating the need for a complete database media restore. The result is a greatly improved SLA, albeit at the cost of doubling the required database storage.

If you are keen on SLA for your VLDB and contemplating moving the Oracle database to a public cloud such as AWS, you could set up a similar database protection structure using resources such as AWS FSx ONTAP for staging your standby database image copy. In this documentation, we demonstrate how to provision and export an NFS file system from AWS FSx ONTAP to be mounted on an Oracle database server for staging a standby database copy for quick recovery in the event of a primary storage failure.

Better yet, we also show how you could leverage NetApp FlexClone to create a copy of the same staging NFS file system for other use cases such as standing up a dev/test Oracle environment with this same standby database image copy without additional storage investment.

This solution addresses the following use cases:

- An Oracle VLDB image copy incremental merge via RMAN on NFS mount point off AWS FSx ONTAP storage.
- Quick recovery of an Oracle VLDB by switching to database image copy on FSx ONTAP storage in the event of failure.
- Clone FSx ONTAP NFS file system volume storing an Oracle VLDB image copy to be used for standing up another database instance for other use cases.

**Audience**

This solution is intended for the following people:

- A DBA who set up Oracle VLDB image copy incremental merge via RMAN in AWS for faster database recovery.
- A database solution architect who tests Oracle workloads in the AWS public cloud.
- A storage administrator who manages Oracle databases deployed to AWS FSx ONTAP storage.
- An application owner who would like to stand up Oracle databases in AWS FSx/EC2 environment.

**Solution test and validation environment**

The testing and validation of this solution was performed in an AWS FSx ONTAP and EC2 environment that might not match the final deployment environment. For more information, see the section [Key Factors for Deployment Consideration].

**Architecture**
Hardware and software components

**Hardware**

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
<th>Consideration</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSx ONTAP storage</td>
<td>Current version offered by AWS</td>
<td>One FSx HA cluster in the same VPC and availability zone</td>
</tr>
<tr>
<td>EC2 instance for compute</td>
<td>t2.xlarge/4vCPU/16G</td>
<td>Two EC2 T2 xlarge EC2 instances, one as primary DB server and the other as a clone DB server</td>
</tr>
</tbody>
</table>

**Software**

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
<th>Consideration</th>
</tr>
</thead>
<tbody>
<tr>
<td>RedHat Linux</td>
<td>RHEL-8.6.0_HVM-20220503-x86_64-2-Hourly2-GP2</td>
<td>Deployed RedHat subscription for testing</td>
</tr>
<tr>
<td>Oracle Grid Infrastructure</td>
<td>Version 19.18</td>
<td>Applied RU patch p34762026_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle Database</td>
<td>Version 19.18</td>
<td>Applied RU patch p34765931_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle OPatch</td>
<td>Version 12.2.0.1.36</td>
<td>Latest patch p6880880_190000_Linux-x86-64.zip</td>
</tr>
</tbody>
</table>

**Key factors for deployment consideration**

- **Oracle VLDB storage layout for RMAN incremental merge.** In our tests and validations, the NFS volume for Oracle incremental backup and merge is allocated from a single FSx file system, which has 4GBps throughput, 160,000 raw SSD IOPS, and 192TiB capacity limit. For deployment over the thresholds,
multiple FSx file systems can be concatenated in parallel with multiple NFS mount points to provide higher capacity.

- **Oracle recoverability using RMAN incremental merge.** The RMAN incremental backup and merge is generally executed at user defined frequency based on your RTO and RPO objectives. If there are total loss of primary data storage and/or archived logs, the data loss can occur. The Oracle database can be recovered up to last incremental backup that is available from FSx database backup image copy. To minimize the data loss, Oracle flash recovery area can be setup on FSx NFS mount point and archived logs are backed up to FSx NFS mount along with database image copy.

- **Running Oracle VLDB off FSx NFS file system.** Unlike other bulk storage for database backup, AWS FSx ONTAP is a cloud enabled production grade storage that delivers high level of performance and storage efficiency. Once Oracle VLDB switches over from primary storage to image copy on FSx ONTAP NFS file system, database performance can be maintained at high level while the primary storage failure is addressed. You can take comfort to know that user application experience does not suffer as the result of primary storage failure.

- **FlexClone Oracle VLDB image copy of NFS volume for other use cases.** AWS FSx ONTAP FlexClone provides shared copies of the same NFS data volume that are writable. Thus, they can be used for many other use cases while still maintaining the integrity of staging Oracle VLDB image copy even when Oracle database is switched over. This provides tremendous storage cost saving by substantially reducing VLDB storage footprint. NetApp recommends to minimize FlexClone activities in the event of database switching over from primary storage to database image copy in order to maintain Oracle performance at high level.

- **EC2 compute instances.** In these tests and validations, we used an AWS EC2 t2.xlarge instance as the Oracle database compute instance. NetApp recommends using an M5 type EC2 instance as the compute instance for Oracle in production deployment because it is optimized for database workload. You need to size the EC2 instance appropriately for the number of vCPUs and the amount of RAM based on actual workload requirements.

- **FSx storage HA clusters single- or multi-zone deployment.** In these tests and validations, we deployed an FSx HA cluster in a single AWS availability zone. For production deployment, NetApp recommends deploying an FSx HA pair in two different availability zones. An FSx HA cluster is always provisioned in a HA pair that is sync mirrored in a pair of active-passive file systems to provide storage-level redundancy. Multi-zone deployment further enhances high availability in the event of failure in a single AWS zone.

- **FSx storage cluster sizing.** An Amazon FSx for ONTAP storage file system provides up to 160,000 raw SSD IOPS, up to 4GBps throughput, and a maximum of 192TiB capacity. However, you can size the cluster in terms of provisioned IOPS, throughput, and the storage limit (minimum 1,024 GiB) based on your actual requirements at the time of deployment. The capacity can be adjusted dynamically on the fly without affecting application availability.

- **dNFS configuration.** dNFS is built into Oracle kernel and is known to dramatically increase Oracle database performance when Oracle is deployed to NFS storage. dNFS is packaged into Oracle binary but is not turned on by default. It should be turned on for any Oracle database deployment on NFS. For multiple FSx file systems deployment for a VLDB, dNFS multi-path to different FSx NFS file systems should be properly configured.

### Solution deployment

It is assumed that you already have your Oracle VLDB deployed in AWS EC2 environment within a VPC. If you need help on Oracle deployment in AWS, please refer to following technical reports for help.

- **Oracle Database Deployment on EC2 and FSx Best Practices**
- **Oracle Database Deployment and Protection in AWS FSx/EC2 with iSCSI/ASM**
- **Oracle 19c in Standalone Restart on AWS FSx/EC2 with NFS/ASM**
Your Oracle VLDB can be running either on a FSx ONTAP or any other storage of choices within the AWS EC2 ecosystem. The following section provides step-by-step deployment procedures for setting up RMAN incremental merge to an image copy of an Oracle VLDB that is staging in an NFS mount off AWS FSx ONTAP storage.

**Prerequisites for deployment**

Deployment requires the following prerequisites.

1. An AWS account has been set up, and the necessary VPC and network segments have been created within your AWS account.

2. From the AWS EC2 console, you must deploy two EC2 Linux instances, one as the primary Oracle DB server and an optional alternative clone target DB server. See the architecture diagram in the previous section for more details about the environment setup. Also review the User Guide for Linux instances for more information.

3. From the AWS EC2 console, deploy Amazon FSx for ONTAP storage HA clusters to host the NFS volumes that stores the Oracle database standby image copy. If you are not familiar with the deployment of FSx storage, see the documentation Creating FSx for ONTAP file systems for step-by-step instructions.

4. Steps 2 and 3 can be performed using the following Terraform automation toolkit, which creates an EC2 instance named `ora_01` and an FSx file system named `fsx_01`. Review the instruction carefully and change the variables to suit your environment before execution. The template can be easily revised for your own deployment requirements.

   ```bash
   git clone https://github.com/NetApp-Automation/na_aws_fsx_ec2_deploy.git
   ```

   Ensure that you have allocated at least 50G in EC2 instance root volume in order to have sufficient space to stage Oracle installation files.

**Provision and export NFS volume to be mounted to EC2 DB instance host**
In this demonstration, we will show how to provision an NFS volume from the command line by login to an FSx cluster via ssh as fsxadmin user through FSx cluster management IP. Alternatively, the volume can be allocated using the AWS FSx console as well. Repeat the procedures on other FSx file systems if more than one FSx file system are set up to accommodate the size of the database.

1. First, provision NFS volume via CLI by logging to the FSx cluster through SSH as the fsxadmin user. Change to your FSx cluster management IP address, which can be retrieved from AWS FSx ONTAP UI console.

   ```
   ssh fsxadmin@172.30.15.53
   ```

2. Create NFS volume the same size as your primary storage for storing primary Oracle VLDB database data files image copy.

   ```
   vol create -volume ora_01_copy -aggregate aggr1 -size 100G -state online -type RW -junction-path /ora_01_copy -snapshot-policy none -tiering-policy snapshot-only
   ```

3. Alternatively, the volume can be provisioned from AWS FSx console UI with options: storage efficiency Enabled, security style Unix, Snapshot policy None, and Storage tiering Snapshot Only as show below.
4. Create a customized snapshot policy for oracle database with a daily schedule and 30 days retention. You should adjust the policy to fit your specific needs in terms of snapshot frequency and retention window.

```
snapshot policy create -policy oracle -enabled true -schedule1 daily -count1 30
```

Apply policy to provisioned NFS volume for RMAN incremental backup and merge.

```
vol modify -volume ora_01_copy -snapshot-policy oracle
```

5. Login to EC2 instance as ec2-user and create a directory /nfsfsxn. Create additional mount point directories for additional FSx file systems.

```
sudo mkdir /nfsfsxn
```

6. Mount the FSx ONTAP NFS volume to EC2 DB instance host. Change to your FSx virtual server NFS lif address. The NFS lif address can be retrieved from FSx ONTAP UI console.
7. Change mount point ownership to oracle:oinstall, change to your oracle user name and primary group as necessary.

```
sudo chown oracle:oinstall /nfsfsxn
```

Setup Oracle RMAN incremental merge to image copy on FSx
RMAN incremental merge update the staging database data files image copy continuously at every incremental backup/merge interval. The image copy of database backup will be as up to date as the frequency you execute the incremental backup/merge. So, take into consideration of database performance, your RTO and RPO objectives when deciding the frequency of RMAN incremental backup and merge.

1. Login to primary DB server EC2 instance as oracle user
2. Create an oracopy directory under mount point /nfsfsxn to store oracle data files image copies and archlog directory for Oracle flash recovery area.

```bash
mkdir /nfsfsxn/oracopy
```  

```bash
mkdir /nfsfsxn/archlog
```  

3. Login to Oracle database via sqlplus, enable block change tracking for faster incremental backup and change Oracle flash recovery area to FSxN mount if it is currently on primary storage. This allows the RMAN default control file/spfile autobackup and archived logs to be backed up to FSxN NFS mount for recovery.

```sql
sqlplus / as sysdba
```

From sqlplus prompt, execute following command.

```sql
alter database enable block change tracking using file /
    /nfsfsxn/oracopy/bct_db1.ctf'
```  

```sql
alter system set db_recovery_file_dest='/nfsfsxn/archlog/
    scope=both;
```  

4. Create a RMAN backup and incremental merge script. The script allocates multiple channels for parallel RMAN backup and merge. First execution would generate the initial full baseline image copy. In a complete run, it first purges obsolete backups that are outside of retention window to keep staging area clean. It then switches current log file before merge and backup. The incremental backup follows the merge so that the database image copy is trailing current database state by one backup/merge cycle. The merge and backup order can be reversed for quicker recovery at user’s preference. The RMAN script can be integrated into a simple shell script to be executed from crontab on the primary DB server. Ensure control file autobackup is on in RMAN setting.
vi /home/oracle/rman_bkup_merge.cmd

Add following lines:

RUN
{
    allocate channel c1 device type disk format '/nfsfsxn/oracopy/%U';
    allocate channel c2 device type disk format '/nfsfsxn/oracopy/%U';
    allocate channel c3 device type disk format '/nfsfsxn/oracopy/%U';
    allocate channel c4 device type disk format '/nfsfsxn/oracopy/%U';
    delete obsolete;
    sql 'alter system archive log current';
    recover copy of database with tag 'OraCopyBKUPonFSxN_level_0';
    backup incremental level 1 copies=1 for recover of copy with tag 'OraCopyBKUPonFSxN_level_0' database;
}

5. At EC2 DB server, login to RMAN locally as oracle user with or without RMAN catalog. In this demonstration, we are not connecting to a RMAN catalog.

    rman target / nocatalog;

    output:

    [oracle@ip-172-30-15-99 ~]$ rman target / nocatalog;

    Recovery Manager: Release 19.0.0.0.0 - Production on Wed May 24 17:44:49 2023
    Version 19.18.0.0.0
    Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.

    connected to target database: DB1 (DBID=1730530050)
    using target database control file instead of recovery catalog

    RMAN>

6. From RMAN prompt, execute the script. First execution creates a baseline database image copy and subsequent executions merge and update the baseline image copy incrementally. The following is how to execute the script and the typical output. Set the number of channels to match the CPU cores on the host.

    RMAN> @/home/oracle/rman_bkup_merge.cmd
RUN

allocate channel c1 device type disk format '/nfsfsxn/oracopy/%U';
allocate channel c2 device type disk format '/nfsfsxn/oracopy/%U';
allocate channel c3 device type disk format '/nfsfsxn/oracopy/%U';
allocate channel c4 device type disk format '/nfsfsxn/oracopy/%U';
delete obsolete;
sql 'alter system archive log current';
recover copy of database with tag 'OraCopyBKUPonFSxN_level_0';
backup incremental level 1 copies=1 for recover of copy with tag 'OraCopyBKUPonFSxN_level_0' database;
}

allocated channel: c1
channel c1: SID=411 device type=DISK

allocated channel: c2
channel c2: SID=146 device type=DISK

allocated channel: c3
channel c3: SID=402 device type=DISK

allocated channel: c4
channel c4: SID=37 device type=DISK

Starting recover at 17-MAY-23
no copy of datafile 1 found to recover
no copy of datafile 3 found to recover
no copy of datafile 4 found to recover
no copy of datafile 5 found to recover
no copy of datafile 6 found to recover
no copy of datafile 7 found to recover.

Finished recover at 17-MAY-23

Starting backup at 17-MAY-23
channel c1: starting incremental level 1 datafile backup set
channel c1: specifying datafile(s) in backup set
input datafile file number=00022
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.287.1137018311
input datafile file number=00026
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.291.1137018481
input datafile file number=00030
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.295.1137018787
input datafile file number=00011
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/undotbs1.271.113668041
input datafile file number=00035
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.300.1137019181
channel c1: starting piece 1 at 17-MAY-23
channel c2: starting incremental level 1 datafile backup set
channel c2: specifying datafile(s) in backup set
input datafile file number=00023
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.288.1137018359
input datafile file number=00027
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.292.1137018523
input datafile file number=00030
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.296.1137018837
input datafile file number=00009
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/system.272.113668041
input datafile file number=00034
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.299.1137019117
.
.
Finished backup at 17-MAY-23

Starting Control File and SPFILE Autobackup at 17-MAY-23
piece
handle=+LOGS/DB1/AUTOBACKUP/2023_05_17/s_1137095435.367.1137095435
comment=NONE
Finished Control File and SPFILE Autobackup at 17-MAY-23
released channel: c1
released channel: c2
released channel: c3
released channel: c4

RMAN> **end-of-file**
7. List database image copy after backup to observe that a database image copy has been created in FSx ONTAP NFS mount point.

RMAN> list copy of database tag 'OraCopyBKUPonFSxN_level_0';

List of Datafile Copies
=======================

<table>
<thead>
<tr>
<th>Key</th>
<th>File S</th>
<th>Completion Time</th>
<th>Ckp SCN</th>
<th>Ckp Time</th>
<th>Sparse</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>1</td>
<td>A 17-MAY-23</td>
<td>3009819</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-1_0h1sd7ae</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>A 17-MAY-23</td>
<td>3009826</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-3_0i1sd7at</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>4</td>
<td>A 17-MAY-23</td>
<td>3009830</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-4_0j1sd7b4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>5</td>
<td>A 17-MAY-23</td>
<td>2383520</td>
<td>12-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-5_0p1sd7cf</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Container ID: 2, PDB Name: PDB$SEED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>6</td>
<td>A 17-MAY-23</td>
<td>2383520</td>
<td>12-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-6_0o1sd7c8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Container ID: 2, PDB Name: PDB$SEED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>7</td>
<td>A 17-MAY-23</td>
<td>3009907</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-7_101sd7dl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>8</td>
<td>A 17-MAY-23</td>
<td>2383520</td>
<td>12-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-8_0v1sd7di</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID</td>
<td>PDB Name</td>
<td>Date</td>
<td>Time</td>
<td>Size</td>
<td>Status</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------</td>
<td>----------</td>
<td>----------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>28</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-9_0q1sd7cm</td>
<td>17-MAY-23</td>
<td>3009871</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-10_0k1sd7bb</td>
<td>17-MAY-23</td>
<td>3009849</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-11_0n1sd7c1</td>
<td>17-MAY-23</td>
<td>3009862</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-12_111sd7dm</td>
<td>17-MAY-23</td>
<td>3009909</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-13_0rlsd7ct</td>
<td>17-MAY-23</td>
<td>3009876</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-14_01l1sd7bi</td>
<td>17-MAY-23</td>
<td>3009854</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-15_0t1sd7db</td>
<td>17-MAY-23</td>
<td>3009900</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-16_111sd7dm</td>
<td>17-MAY-23</td>
<td>3009911</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Container ID</td>
<td>PDB Name</td>
<td>Date</td>
<td>Time</td>
<td>Status</td>
<td>Name</td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
<td>---------------</td>
<td>---------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>4</td>
<td>DB1_PDB2</td>
<td>17-MAY-23</td>
<td>3009895</td>
<td>NO</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-17_01sd7d4</td>
</tr>
<tr>
<td>5</td>
<td>DB1_PDB3</td>
<td>17-MAY-23</td>
<td>3009858</td>
<td>NO</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-18_01sd7bq</td>
</tr>
<tr>
<td>5</td>
<td>DB1_PDB3</td>
<td>17-MAY-23</td>
<td>3009903</td>
<td>NO</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-19_01sd7de</td>
</tr>
<tr>
<td>5</td>
<td>DB1_PDB3</td>
<td>17-MAY-23</td>
<td>3009914</td>
<td>NO</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-20_01sd7do</td>
</tr>
<tr>
<td>3</td>
<td>DB1_PDB1</td>
<td>17-MAY-23</td>
<td>3009019</td>
<td>NO</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-21_01sd6pv</td>
</tr>
<tr>
<td>3</td>
<td>DB1_PDB1</td>
<td>17-MAY-23</td>
<td>3009419</td>
<td>NO</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-22_01sd6r2</td>
</tr>
<tr>
<td>3</td>
<td>DB1_PDB1</td>
<td>17-MAY-23</td>
<td>3009460</td>
<td>NO</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-23_01sd6s5</td>
</tr>
<tr>
<td>No</td>
<td>Date</td>
<td>Time</td>
<td>Size</td>
<td>Date</td>
<td>Action</td>
</tr>
<tr>
<td>----</td>
<td>-----------</td>
<td>-------</td>
<td>---------</td>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>7</td>
<td>17-MAY-23</td>
<td>00:00</td>
<td>3009473</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td>8</td>
<td>17-MAY-23</td>
<td>00:00</td>
<td>3009502</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td>9</td>
<td>17-MAY-23</td>
<td>00:00</td>
<td>3009548</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td>10</td>
<td>17-MAY-23</td>
<td>00:00</td>
<td>3009576</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td>11</td>
<td>17-MAY-23</td>
<td>00:00</td>
<td>3009590</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td>12</td>
<td>17-MAY-23</td>
<td>00:00</td>
<td>3009619</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td>13</td>
<td>17-MAY-23</td>
<td>00:00</td>
<td>3009648</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td>14</td>
<td>17-MAY-23</td>
<td>00:00</td>
<td>3009671</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
</tbody>
</table>

Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-24_051sd6t9
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-25_061sd6uc
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-26_071sd6vf
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-27_081sd70i
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-28_091sd71l
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-29_0a1sd72o
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-30_0b1sd73r
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-31_0c1sd74u
Tag: ORACOPYBKUPONFSXN_LEVEL_0
8. Report schema from Oracle RMAN command prompt to observe that current active database data files are in primary storage ASM +DATA disk group.

RMAN> report schema;

Report of database schema for database with db_unique_name DB1

List of Permanent Datafiles
=======================================
File Size(MB) Tablespace   RB segs Datafile Name
------------ ----------- -------------- -----------------------------------------------------
 1 1060  SYSTEM     YES +DATA/DB1/DATAFILE/system.257.1136666315
 3  810  SYSAUX     NO +DATA/DB1/DATAFILE/sysaux.258.1136666361
 4  675 UNDOTBS1    YES +DATA/DB1/DATAFILE/undotbs1.259.1136666385
5  400  PDB$SEED:SYSTEM  NO
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/system.266.11366
67165
6  460  PDB$SEED:SYSAUX  NO
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/sysaux.267.11366
67165
7  5  USERS  NO
+DATA/DB1/DATAFILE/users.260.1136666387
8  230  PDB$SEED:UNDOTBS1  NO
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/undotbs1.268.113
6667165
9  400  DB1_PDB1:SYSTEM  YES
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/system.272.11366
68041
10  490  DB1_PDB1:SYSAUX  NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/sysaux.273.11366
68041
11  465  DB1_PDB1:UNDOTBS1  YES
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/undotbs1.271.113
6668041
12  5  DB1_PDB1:USERS  NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/users.275.113666
8057
13  400  DB1_PDB2:SYSTEM  YES
+DATA/DB1/FB867EA89ECF81C0E053630F1EACB901/DATAFILE/system.277.11366
68057
14  470  DB1_PDB2:SYSAUX  NO
+DATA/DB1/FB867EA89ECF81C0E053630F1EACB901/DATAFILE/sysaux.278.11366
68057
15  235  DB1_PDB2:UNDOTBS1  YES
+DATA/DB1/FB867EA89ECF81C0E053630F1EACB901/DATAFILE/undotbs1.271.113
6668057
16  5  DB1_PDB2:USERS  NO
+DATA/DB1/FB867EA89ECF81C0E053630F1EACB901/DATAFILE/users.280.113666
8071
17  400  DB1_PDB3:SYSTEM  YES
+DATA/DB1/FB867F8A4D4F821CE053630F1EAC69CC/DATAFILE/system.282.11366
68073
18  470  DB1_PDB3:SYSAUX  NO
+DATA/DB1/FB867F8A4D4F821CE053630F1EAC69CC/DATAFILE/sysaux.283.11366
68073
19  235  DB1_PDB3:UNDOTBS1  YES
+DATA/DB1/FB867F8A4D4F821CE053630F1EAC69CC/DATAFILE/undotbs1.281.113
6668073
20  5  DB1_PDB3:USERS  NO
+DATA/DB1/FB867F8A4D4F821CE053630F1EAC69CC/DATAFILE/users.285.113666
<table>
<thead>
<tr>
<th>#</th>
<th>Size</th>
<th>Database</th>
<th>Description</th>
<th>Verdict</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>4096</td>
<td>DB1_PDB1:SOE</td>
<td>NO</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.286.11370182</td>
</tr>
<tr>
<td>22</td>
<td>4096</td>
<td>DB1_PDB1:SOE</td>
<td>NO</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.287.11370183</td>
</tr>
<tr>
<td>23</td>
<td>4096</td>
<td>DB1_PDB1:SOE</td>
<td>NO</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.288.11370183</td>
</tr>
<tr>
<td>24</td>
<td>4096</td>
<td>DB1_PDB1:SOE</td>
<td>NO</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.289.11370184</td>
</tr>
<tr>
<td>25</td>
<td>4096</td>
<td>DB1_PDB1:SOE</td>
<td>NO</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.290.11370184</td>
</tr>
<tr>
<td>26</td>
<td>4096</td>
<td>DB1_PDB1:SOE</td>
<td>NO</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.291.11370184</td>
</tr>
<tr>
<td>27</td>
<td>4096</td>
<td>DB1_PDB1:SOE</td>
<td>NO</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.292.11370185</td>
</tr>
<tr>
<td>28</td>
<td>4096</td>
<td>DB1_PDB1:SOE</td>
<td>NO</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.293.11370187</td>
</tr>
<tr>
<td>29</td>
<td>4096</td>
<td>DB1_PDB1:SOE</td>
<td>NO</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.294.11370187</td>
</tr>
<tr>
<td>30</td>
<td>4096</td>
<td>DB1_PDB1:SOE</td>
<td>NO</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.295.11370187</td>
</tr>
<tr>
<td>31</td>
<td>4096</td>
<td>DB1_PDB1:SOE</td>
<td>NO</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.296.11370188</td>
</tr>
<tr>
<td>32</td>
<td>4096</td>
<td>DB1_PDB1:SOE</td>
<td>NO</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.297.11370189</td>
</tr>
<tr>
<td>33</td>
<td>4096</td>
<td>DB1_PDB1:SOE</td>
<td>NO</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.298.11370190</td>
</tr>
<tr>
<td>34</td>
<td>4096</td>
<td>DB1_PDB1:SOE</td>
<td>NO</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.299.11370191</td>
</tr>
<tr>
<td>35</td>
<td>4096</td>
<td>DB1_PDB1:SOE</td>
<td>NO</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.300.11370191</td>
</tr>
</tbody>
</table>
List of Temporary Files
=======================
<table>
<thead>
<tr>
<th>File Size(MB)</th>
<th>Tablespace</th>
<th>Maxsize(MB)</th>
<th>Tempfile Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>123</td>
<td>TEMP</td>
<td>32767</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+DATA/DB1/TEMPFILE/temp.265.113666447</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>123</td>
<td>PDB$SEED:TEMP</td>
<td>32767</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+DATA/DB1/FB864A929AEB79B9E053630F1EAC7046/TEMPFILE/temp.269.1136667 185</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10240</td>
<td>DB1_PDB1:TEMP</td>
<td>32767</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/TEMPFILE/temp.274.1136668 051</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>123</td>
<td>DB1_PDB2:TEMP</td>
<td>32767</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+DATA/DB1/FB867EA89ECF81C0E053630F1EACB901/TEMPFILE/temp.279.1136668 067</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>123</td>
<td>DB1_PDB3:TEMP</td>
<td>32767</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+DATA/DB1/FB867F8A4D4F821CE053630F1EAC69CC/TEMPFILE/temp.284.1136668 081</td>
<td></td>
</tr>
</tbody>
</table>

RMAN>

9. Validate database image copy from OS NFS mount point.

[oracle@ip-172-30-15-99 ~]$ ls -l /nfsfsxn/oracopy/
total 70585148
-rw-r----- 1 oracle asm 4294975488 May 17 18:09 data_D-DB1_I-1730530050_TS-SOE_FNO-21_021sd6pv
-rw-r----- 1 oracle asm 4294975488 May 17 18:10 data_D-DB1_I-1730530050_TS-SOE_FNO-22_031sd6rz2
-rw-r----- 1 oracle asm 4294975488 May 17 18:10 data_D-DB1_I-1730530050_TS-SOE_FNO-23_041sd66s5
-rw-r----- 1 oracle asm 4294975488 May 17 18:11 data_D-DB1_I-1730530050_TS-SOE_FNO-24_051sd6t9
-rw-r----- 1 oracle asm 4294975488 May 17 18:11 data_D-DB1_I-1730530050_TS-SOE_FNO-25_061sd6uc
-rw-r----- 1 oracle asm 4294975488 May 17 18:12 data_D-DB1_I-1730530050_TS-SOE_FNO-26_071sd6vf
-rw-r----- 1 oracle asm 4294975488 May 17 18:13 data_D-DB1_I-1730530050_TS-SOE_FNO-27_081sd70i
-rw-r----- 1 oracle asm 4294975488 May 17 18:13 data_D-DB1_I-1730530050_TS-SOE_FNO-28_091sd71l
-rw-r----- 1 oracle asm 4294975488 May 17 18:14 data_D-DB1_I-1730530050_TS-SOE_FNO-29_0a1sd72o
-rw-r----- 1 oracle asm 4294975488 May 17 18:14 data_D-DB1_I-
This completes the setup of Oracle database standby image copy backup and merge.

Switch Oracle DB to image copy for quick recovery
In the event of a failure due to primary storage issue such as data loss or corruption, database can be quickly switched over to image copy on FSx ONTAP NFS mount and recovered to current state without database restore. Eliminating media restoration speeds up the database recovery tremendously for a VLDB. This use case assumes that the database host instance is intact and database control file, archived and current logs are all available for recovery.

1. Login to EC2 DB server host as oracle user and create a test table before switch over.

```sql
[ec2-user@ip-172-30-15-99 ~]$ sudo su
[root@ip-172-30-15-99 ec2-user]# su - oracle
Last login: Thu May 18 14:22:34 UTC 2023
[oracle@ip-172-30-15-99 ~]$ sqlplus / as sysdb

SQL*Plus: Release 19.0.0.0.0 - Production on Thu May 18 14:30:36 2023
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.18.0.0.0

SQL> show pdb

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>CON_NAME</th>
<th>OPEN MODE</th>
<th>RESTRICTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>READ ONLY</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>DB1_PDB1</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>4</td>
<td>DB1_PDB2</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>5</td>
<td>DB1_PDB3</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
</tbody>
</table>

SQL> alter session set container=db1_pdb1;

Session altered.

SQL> create table test (id integer, dt timestamp, event varchar(100));

Table created.

SQL> insert into test values(1, sysdate, 'test oracle incremental merge switch to copy');

1 row created.
SQL> commit;
Commit complete.

SQL> select * from test;

<table>
<thead>
<tr>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>---</td>
</tr>
<tr>
<td>DT</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>EVENT</td>
</tr>
</tbody>
</table>

---

1
18-MAY-23 02.35.37.000000 PM
test oracle incremental merge switch to copy

SQL>

2. Simulate a failure by shutdown abort database, then start up oracle in mount stage.

SQL> shutdown abort;
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.

Total System Global Area 1.2885E+10 bytes
Fixed Size 9177880 bytes
Variable Size 1778384896 bytes
Database Buffers 1.1073E+10 bytes
Redo Buffers 24375296 bytes
Database mounted.

SQL>

3. As oracle user, connect to Oracle database via RMAN to switch database to copy.

RMAN> switch database to copy;

datafile 1 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-1_0h1sd7ae"
datafile 3 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-3_0i1sd7at"
datafile 4 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-4_0j1sd7b4"
datafile 5 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-5_0p1sd7cf"
datafile 6 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-6_0o1sd7c8"
datafile 7 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-7_101sd7d1"
datafile 8 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-8_0v1sd7di"
datafile 9 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-9_0q1sd7cm"
datafile 10 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-10_0k1sd7bb"
datafile 11 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-11_0n1sd7c1"
datafile 12 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-12_111sd7dm"
datafile 13 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-13_0r1sd7ct"
datafile 14 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-14_0l1sd7bi"
datafile 15 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-15_0t1sd7db"
datafile 16 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-16_121sd7dn"
datafile 17 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-17_0s1sd7d4"
datafile 18 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-18_0m1sd7bq"
datafile 19 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-19_0u1sd7de"
datafile 20 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-20_131sd7do"
datafile 21 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-21_021sd6pv"
datafile 22 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-22_031sd6r2"
datafile 23 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-23_041sd6s5"
datafile 24 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-24_051sd6t9"
datafile 25 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-25_061sd6uc"
datafile 26 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-26_071sd6vf"
Datafile 27 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-27_081sd70i"
datafile 28 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-28_091sd711"
datafile 29 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-29_0a1sd72o"
datafile 30 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-30_0b1sd73r"
datafile 31 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-31_0c1sd74u"
datafile 32 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-32_0d1sd762"
datafile 33 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-33_0e1sd775"
datafile 34 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-34_0f1sd788"
datafile 35 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-35_0g1sd79b"

4. Recover and open database to bring it up to current from last incremental backup.

RMAN> recover database;

Starting recover at 18-MAY-23
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=392 device type=DISK
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
destination for restore of datafile 00009: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-9_0q1sd7cm
destination for restore of datafile 00023: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-23_041sd6s5
destination for restore of datafile 00027: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-27_081sd70i
destination for restore of datafile 00031: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-31_0c1sd74u
destination for restore of datafile 00034: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-34_0f1sd788
channel ORA_DISK_1: reading from backup piece
/nfsfsxn/oracopy/321sfous_98_1_1
channel ORA_DISK_1: piece handle=/nfsfsxn/oracopy/321sfous_98_1_1
tag=ORACOPYBKUPONFSXN_LEVEL_0
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
destination for restore of datafile 00010: /nfsfsxn/oracopy/data_DB1_I-1730530050_TS-SYSAUX_FNO-10_0k1sd7bb
destination for restore of datafile 00021: /nfsfsxn/oracopy/data_DB1_I-1730530050_TS-SOE_FNO-21_021sd6pv
destination for restore of datafile 00025: /nfsfsxn/oracopy/data_DB1_I-1730530050_TS-SOE_FNO-25_061sd6uc
.
.
.
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
destination for restore of datafile 00016: /nfsfsxn/oracopy/data_DB1_I-1730530050_TS-USERS_FNO-16_121sd7dn
channel ORA_DISK_1: reading from backup piece /nfsfsxn/oracopy/3ilsfov0_114_1_1
channel ORA_DISK_1: piece handle=/nfsfsxn/oracopy/3ilsfov0_114_1_1
  tag=ORACOPYBKUPONFSXN_LEVEL_0
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
destination for restore of datafile 00020: /nfsfsxn/oracopy/data_DB1_I-1730530050_TS-USERS_FNO-20_131sd7do
channel ORA_DISK_1: reading from backup piece /nfsfsxn/oracopy/3jlsfov0_115_1_1
channel ORA_DISK_1: piece handle=/nfsfsxn/oracopy/3jlsfov0_115_1_1
  tag=ORACOPYBKUPONFSXN_LEVEL_0
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01

starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 18-MAY-23

RMAN> alter database open;

Statement processed

RMAN>
5. Check database structure from sqlplus after recovery to observe that all database data files with exception of control, temp, and current log files are now switched over to copy on FSx ONTAP NFS file system.

```sql
SQL> select name from v$datafile
    2  union
    3  select name from v$tempfile
    4  union
    5  select name from v$controlfile
    6  union
    7  select member from v$logfile;

NAME
---------------------------------------------
+DATA/DB1/CONTROLFILE/current.261.1136666435
+DATA/DB1/FB864A929AEB79B9E053630F1EAC7046/TEMPFILE/temp.269.1136667185
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/TEMPFILE/temp.274.1136668051
+DATA/DB1/FB867EA89ECF81C0E053630F1EACB901/TEMPFILE/temp.279.1136668067
+DATA/DB1/FB867F8A4D4F821CE053630F1EAC69CC/TEMPFILE/temp.284.1136668081
+DATA/DB1/ONLINELOG/group_1.262.1136666437
+DATA/DB1/ONLINELOG/group_2.263.1136666437
+DATA/DB1/ONLINELOG/group_3.264.1136666437
+DATA/DB1/TEMPFILE/temp.265.1136666447
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-21_021sd6pv
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-22_031sd6r2

NAME
---------------------------------------------
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-23_041sd6s5
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-24_051sd6t9
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-25_061sd6uc
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-26_071sd6vf
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-27_081sd70i
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-28_091sd71l
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-29_0a1sd72o
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-30_0b1sd73r
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-31_0c1sd74u
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-32_0d1sd762
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-33_0e1sd775
```
6. From SQL plus, check the content of test table we have inserted before the switch over to copy
SQL> show pdb

CON_ID  CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
 2  PDB$SEED                       READ ONLY  NO
 3  DB1_PDB1                       READ WRITE NO
 4  DB1_PDB2                       READ WRITE NO
 5  DB1_PDB3                       READ WRITE NO

SQL> alter session set container=db1_pdb1;

Session altered.

SQL> select * from test;

<table>
<thead>
<tr>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>DT</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>EVENT</td>
</tr>
</tbody>
</table>

1
18-MAY-23 02.35.37.000000 PM
test oracle incremental merge switch to copy

7. You could run the Oracle database in FSx NFS mount for an extended period without a performance penalty because FSx ONTAP is redundant production-grade storage that delivers high performance. When the primary storage issue is fixed, you can swing back to it by reversing the incremental backup merge processes with minimal downtime.

Oracle DB recovery from image copy to different EC2 DB instance host
In a failure when both primary storage and EC2 DB instance host are lost, the recovery can not be conducted from the original server. Fortunately, you still have an Oracle database backup image copy on the redundant FSxN NFS file system. You could quickly provision another identical EC2 DB instance and easily mount the image copy of your VLDB to the new EC2 DB host via NFS to run recovery. In this section, we will demonstrate the step-by-step procedures for doing so.

1. Insert a row to test table we have created previously for Oracle database restoring to alternative host validation.

```sql
[oracle@ip-172-30-15-99 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Tue May 30 17:21:05 2023
Copyright (c) 1982, 2022, Oracle.  All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.18.0.0.0

SQL> show pdbs

CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
 2 PDB$SEED                       READ ONLY  NO
 3 DB1_PDB1                       READ WRITE NO
 4 DB1_PDB2                       READ WRITE NO
 5 DB1_PDB3                       READ WRITE NO

SQL> alter session set container=db1_pdb1;

Session altered.

SQL> insert into test values(2, sysdate, 'test recovery on a new EC2 instance host with image copy on FSxN');

1 row created.

SQL> commit;

Commit complete.

SQL> select * from test;

  1 | 2023-05-30 | test recovery on a new EC2 instance host with image copy on FSxN |
---+-----------+---------------------------------------------------------------
  1 | 2023-05-30 | test recovery on a new EC2 instance host with image copy on FSxN |
2. As oracle user, run RMAN incremental backup and merge to flush the transaction to backup set on FSxN NFS mount.

```
[oracle@ip-172-30-15-99 ~]$ rman target / nocatalog

Recovery Manager: Release 19.0.0.0.0 - Production on Tue May 30 17:26:03 2023
Version 19.18.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.

connected to target database: DB1 (DBID=1730530050)
using target database control file instead of recovery catalog

RMAN> @rman_bkup_merge.cmd
```

3. Shutdown primary EC2 DB instance host to simulate a total failure of storage and DB server host.
4. Provision a new EC2 DB instance host ora_02 with same OS and version via AWS EC2 console. Configure OS kernal with same patches as primary EC2 DB server host, Oracle preinstall RPM, and add swap space to the host as well. Install same version and patches of Oracle as in primary EC2 DB server host with software only option. These tasks can be automated with NetApp automation toolkit as available from below links.

**Toolkit:** na_oracle19c_deploy
**Documentation:** Automated Deployment of Oracle19c for ONTAP on NFS
5. Configure oracle environment similarly to primary EC2 DB instance host ora_01, such as oratab, oralinst.loc, and oracle user .bash_profile. It is a good practice to backup those files to FSxN NFS mount point.

6. The Oracle database backup image copy on FSxN NFS mount is stored on a FSx cluster that spans AWS availability zones for redundancy, high availability, and high performance. The NFS file system can be easily mounted to a new server as far as the networking is reachable. The following procedures mount the image copy of an Oracle VLDB backup to newly provisioned EC2 DB instance host for recovery.

As ec2-user, create the mount point.

```
sudo mkdir /nfsfsxn
```

As ec2-user, mount the NFS volume that stored Oracle VLDB backup image copy.

```
sudo mount 172.30.15.19:/ora_01_copy /nfsfsxn -o rw, bg, hard, vers=3, proto=tcp, timeo=600, rsize=262144, wsize=262144, nointr
```

7. Validate the Oracle database backup image copy on FSxN NFS mount point.

```
[ec2-user@ip-172-30-15-124 ~]$ ls -ltr /nfsfsxn/oracopy
```

```text
total 78940700
-rw-r----- 1 oracle 54331  482353152 May 26 18:45 data_D-DB1_I-1730530050_TS-SYSAUX_FNO-6_4m1t508t
-rw-r----- 1 oracle 54331  419438592 May 26 18:45 data_D-DB1_I-1730530050_TS-SYSTEM_FNO-5_4q1t509n
-rw-r----- 1 oracle 54331  241180672 May 26 18:45 data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-8_4t1t50a6
-rw-r----- 1 oracle 54331  450560 May 30 15:29 6b1tf6b8_203_1_1
-rw-r----- 1 oracle 54331  663552 May 30 15:29 6c1tf6b8_204_1_1
-rw-r----- 1 oracle 54331  122880 May 30 15:29 6d1tf6b8_205_1_1
-rw-r----- 1 oracle 54331  507904 May 30 15:29 6e1tf6b8_206_1_1
-rw-r----- 1 oracle 54331  4259840 May 30 15:29 6f1tf6b9_207_1_1
-rw-r----- 1 oracle 54331  9060352 May 30 15:29 6h1tf6b9_209_1_1
-rw-r----- 1 oracle 54331  442368 May 30 15:29 6i1tf6b9_210_1_1
-rw-r----- 1 oracle 54331  475136 May 30 15:29 6j1tf6bb_211_1_1
-rw-r----- 1 oracle 54331  48660480 May 30 15:29 6l1tf6bb_213_1_1
-rw-r----- 1 oracle 54331  589824 May 30 15:29 6m1tf6bb_213_1_1
-rw-r----- 1 oracle 54331  606208 May 30 15:29 6n1tf6bb_214_1_1
-rw-r----- 1 oracle 54331  36840 May 30 15:29 6o1tf6bb_216_1_1
-rw-r----- 1 oracle 54331  36840 May 30 15:29 6p1tf6bc_217_1_1
-rw-r----- 1 oracle 54331  57344 May 30 15:29 6q1tf6bc_219_1_1
-rw-r----- 1 oracle 54331  57344 May 30 15:29 6r1tf6bc_220_1_1
-rw-r----- 1 oracle 54331  57344 May 30 15:29 6t1tf6bc_221_1_1
```
8. Verify the available Oracle archived logs on the FSxN NFS mount for recovery and note the last log file log sequency number. In this case, it is 175. Our recovery point is up to log sequency number 176.

[ec2-user@ip-172-30-15-124 ~]$ ls -ltr
/nfsfsxn/archlog/DB1/archivelog/2023_05_30
 total 5714400
-r-r-r---. 1 oracle 54331 321024 May 30 14:59 o1_mf_1_140__003t9mvn_.arc
-r-r-r---. 1 oracle 54331 48996352 May 30 15:29 o1_mf_1_141__01t9qf6r_.arc
-r-r-r---. 1 oracle 54331 167477248 May 30 15:44 o1_mf_1_142__02t9z90_.arc
-r-r-r---. 1 oracle 54331 1874903040 May 30 17:54 db1_ctl.sql
-r-r-r---. 1 oracle 54331 11091968 May 30 17:32 721tfd6b_226_1_1
-r-r-r---. 1 oracle 54331 2070708224 May 30 17:33 6v1tfd6a_223_1_1
-r-r-r---. 1 oracle 54331 376832 May 30 17:33 7a1tfdit_234_1_1
-r-r-r---. 1 oracle 54331 319488 May 30 17:33 7d1tfdiv_237_1_1
-r-r-r---. 1 oracle 54331 321024 May 30 14:59 o1_mf_1_140__003t9mvn_.arc
-r-r-r---. 1 oracle 54331 48996352 May 30 15:29 o1_mf_1_141__01t9qf6r_.arc
-r-r-r---. 1 oracle 54331 167477248 May 30 15:44 o1_mf_1_142__02t9z90_.arc
-r-r-r---. 1 oracle 54331 1874903040 May 30 17:54 db1_ctl.sql
-r-r-r---. 1 oracle 54331 11091968 May 30 17:32 721tfd6b_226_1_1
-r-r-r---. 1 oracle 54331 2070708224 May 30 17:33 6v1tfd6a_223_1_1
-r-r-r---. 1 oracle 54331 376832 May 30 17:33 7a1tfdit_234_1_1
-r-r-r---. 1 oracle 54331 319488 May 30 17:33 7d1tfdiv_237_1_1
-r-r-r---. 1 oracle 54331 321024 May 30 14:59 o1_mf_1_140__003t9mvn_.arc
-r-r-r---. 1 oracle 54331 48996352 May 30 15:29 o1_mf_1_141__01t9qf6r_.arc
-r-r-r---. 1 oracle 54331 167477248 May 30 15:44 o1_mf_1_142__02t9z90_.arc
-r-r-r---. 1 oracle 54331 1874903040 May 30 17:54 db1_ctl.sql
-r-r-r---. 1 oracle 54331 11091968 May 30 17:32 721tfd6b_226_1_1
-r-r-r---. 1 oracle 54331 2070708224 May 30 17:33 6v1tfd6a_223_1_1
-r-r-r---. 1 oracle 54331 376832 May 30 17:33 7a1tfdit_234_1_1
-r-r-r---. 1 oracle 54331 319488 May 30 17:33 7d1tfdiv_237_1_1
-r-r-r---. 1 oracle 54331 321024 May 30 14:59 o1_mf_1_140__003t9mvn_.arc
-r-r-r---. 1 oracle 54331 48996352 May 30 15:29 o1_mf_1_141__01t9qf6r_.arc
-r-r-r---. 1 oracle 54331 167477248 May 30 15:44 o1_mf_1_142__02t9z90_.arc
-r-r-r---. 1 oracle 54331 1874903040 May 30 17:54 db1_ctl.sql

9. As oracle user, set ORACLE_HOME variable to current Oracle installation on new EC2 instance DB host ora_02, ORACLE_SID to primary Oracle instance SID. In this case, it is db1.

10. As oracle user, create a generic Oracle init file in $ORACLE_HOME/dbs directory with proper admin directories configured. Most importantly, have Oracle flash recovery area point to FSxNFS mount path as defined in primary Oracle VLDB instance. flash recovery area configuration is demonstrated in section Setup Oracle RMAN incremental merge to image copy on FSx. Set the Oracle control file to FSx ONTAP NFS file system.

```bash
vi $ORACLE_HOME/dbs/initdb1.ora
```

With following example entries:
*.audit_file_dest='/u01/app/oracle/admin/db1/adump'
*.audit_trail='db'
*.compatible='19.0.0'
*.control_files=('/nfsfsxn/oracopy/db1.ctl')
*.db_block_size=8192
*.db_create_file_dest='/nfsfsxn/oracopy/
*.db_domain='demo.netapp.com'
*.db_name='db1'
*.db_recovery_file_dest_size=85899345920
*.db_recovery_file_dest='/nfsfsxn/archlog/'
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=db1XDB)' 
*.enable_pluggable_database=true
*.local_listener='LISTENER'
*.nls_language='AMERICAN'
*.nls_territory='AMERICA'
*.open_cursors=300
*.pga_aggregate_target=1024m
*.processes=320
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=10240m
*.undo_tablespace='UNDOTBS1'

The above init file should be replaced by restored backup init file from primary Oracle DB server in the case of discrepancy.

11. As oracle user, launch RMAN to run Oracle recovery on a new EC2 DB instance host.
[oracle@ip-172-30-15-124 dbs]$ rman target / nocatalog;

Recovery Manager: Release 19.0.0.0.0 - Production on Wed May 31 00:56:07 2023
Version 19.18.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.

connected to target database (not started)

RMAN> startup nomount;

Oracle instance started

Total System Global Area   12884900632 bytes

Fixed Size                     9177880 bytes
Variable Size               1778384896 bytes
Database Buffers           11072962560 bytes
Redo Buffers                  24375296 bytes

12. Set database ID. The database ID can be retrieved from Oracle file name of image copy on FSx NFS mount point.

RMAN> set dbid = 1730530050;

executing command: SET DBID

13. Restore controlfile from autobackup. If Oracle controlfile and spfile autobackup is enabled, they are backed up in every incremental backup and merge cycle. The latest backup will be restored if multiple copies are available.
RMAN> restore controlfile from autobackup;

Starting restore at 31-MAY-23
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=2 device type=DISK

recovery area destination: /nfsfsxn/archlog
database name (or database unique name) used for search: DB1
channel ORA_DISK_1: AUTOBACKUP
/nfsfsxn/archlog/DB1/autobackup/2023_05_30/o1_mf_s_1138210401__08qlx
rrr_.bkp found in the recovery area
channel ORA_DISK_1: looking for AUTOBACKUP on day: 20230531
channel ORA_DISK_1: looking for AUTOBACKUP on day: 20230530
channel ORA_DISK_1: restoring control file from AUTOBACKUP
/nfsfsxn/archlog/DB1/autobackup/2023_05_30/o1_mf_s_1138210401__08qlx
rrr_.bkp
channel ORA_DISK_1: control file restore from AUTOBACKUP complete
output file name=/nfsfsxn/oracopy/db1.ctl
Finished restore at 31-MAY-23

14. Restore init file from spfile to a /tmp folder for updating parameter file later to match with primary DB instance.

RMAN> restore spfile to pfile '/tmp/archive/initdb1.ora' from autobackup;

Starting restore at 31-MAY-23
using channel ORA_DISK_1

recovery area destination: /nfsfsxn/archlog
database name (or database unique name) used for search: DB1
channel ORA_DISK_1: AUTOBACKUP
/nfsfsxn/archlog/DB1/autobackup/2023_05_30/o1_mf_s_1138210401__08qlx
rrr_.bkp found in the recovery area
channel ORA_DISK_1: looking for AUTOBACKUP on day: 20230531
channel ORA_DISK_1: looking for AUTOBACKUP on day: 20230530
channel ORA_DISK_1: restoring spfile from AUTOBACKUP
/nfsfsxn/archlog/DB1/autobackup/2023_05_30/o1_mf_s_1138210401__08qlx
rrr_.bkp
channel ORA_DISK_1: SPFILE restore from AUTOBACKUP complete
Finished restore at 31-MAY-23

15. Mount control file and validate the database backup image copy.
RMAN> alter database mount;

released channel: ORA_DISK_1
Statement processed

RMAN> list copy of database tag 'OraCopyBKUPonFSxN_level_0';

List of Datafile Copies
=======================================

<table>
<thead>
<tr>
<th>Key</th>
<th>File S</th>
<th>Completion Time</th>
<th>Ckp SCN</th>
<th>Ckp Time</th>
<th>Sparse</th>
</tr>
</thead>
<tbody>
<tr>
<td>316</td>
<td>1 A</td>
<td>30-MAY-23</td>
<td>4120170</td>
<td>30-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS- SYSTEM_FNO-1_4f1t506m</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>322</td>
<td>3 A</td>
<td>30-MAY-23</td>
<td>4120175</td>
<td>30-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS- SYSAUX_FNO-3_4g1t506m</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>317</td>
<td>4 A</td>
<td>30-MAY-23</td>
<td>4120179</td>
<td>30-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS- UNDOTBS1_FNO-4_4h1t5083</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>221</td>
<td>5 A</td>
<td>26-MAY-23</td>
<td>2383520</td>
<td>12-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS- SYSTEM_FNO-5_4q1t509n</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Container ID: 2, PDB Name: PDB$SEED |
| 216 | 6 A     | 26-MAY-23       | 2383520 | 12-MAY-23       | NO     |
|     | Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS- SYSAUX_FNO-6_4mlt508t |
|     | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
|     | Container ID: 2, PDB Name: PDB$SEED |
| 323 | 7 A     | 30-MAY-23       | 4120207 | 30-MAY-23       | NO     |
|     | Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-7_4ult50a6 |
|     | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
| 227 | 8 A     | 26-MAY-23       | 2383520 | 12-MAY-23       | NO     |
|     | Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS- UNDOTBS1_FNO-8_4t1t50a6 |
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 2, PDB Name: PDB$SEED

308    9    A 30-MAY-23       4120158    30-MAY-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-9_4n1t509m
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

307    10   A 30-MAY-23       4120166    30-MAY-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-10_4ilt5083
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

313    11   A 30-MAY-23       4120154    30-MAY-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-11_4l1t508t
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

315    12   A 30-MAY-23       4120162    30-MAY-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-12_4v1t50aa
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

319    13   A 30-MAY-23       4120191    30-MAY-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-13_4o1t509m
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 4, PDB Name: DB1_PDB2

318    14   A 30-MAY-23       4120183    30-MAY-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-14_4j1t508s
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 4, PDB Name: DB1_PDB2

324    15   A 30-MAY-23       4120199    30-MAY-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-15_4r1t50a6
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 4, PDB Name: DB1_PDB2

325    16   A 30-MAY-23       4120211    30-MAY-23       NO
<p>| Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-16_501t50ad | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
| Container ID: 4, PDB Name: DB1_PDB2 |
| 320 | 17 | A | 30-MAY-23 | 4120195 | 30-MAY-23 | NO |
| Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-17_4p1t509m | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
| Container ID: 5, PDB Name: DB1_PDB3 |
| 321 | 18 | A | 30-MAY-23 | 4120187 | 30-MAY-23 | NO |
| Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-18_4k1t508t | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
| Container ID: 5, PDB Name: DB1_PDB3 |
| 326 | 19 | A | 30-MAY-23 | 4120203 | 30-MAY-23 | NO |
| Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-19_4s1t50a6 | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
| Container ID: 5, PDB Name: DB1_PDB3 |
| 327 | 20 | A | 30-MAY-23 | 4120216 | 30-MAY-23 | NO |
| Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-20_511t50ad | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
| Container ID: 5, PDB Name: DB1_PDB3 |
| 298 | 21 | A | 30-MAY-23 | 4120166 | 30-MAY-23 | NO |
| Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-21_3o1t4ut2 | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
| Container ID: 3, PDB Name: DB1_PDB1 |
| 302 | 22 | A | 30-MAY-23 | 4120154 | 30-MAY-23 | NO |
| Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-22_3p1t4ut3 | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
| Container ID: 3, PDB Name: DB1_PDB1 |
| 297 | 23 | A | 30-MAY-23 | 4120158 | 30-MAY-23 | NO |
| Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-23_3q1t4ut3 | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
| Container ID: 3, PDB Name: DB1_PDB1 |</p>
<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Tag</th>
<th>Container ID</th>
<th>PDB Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>306</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-24_3r1t4ut3</td>
<td>ORACOPYBKUPONFSXN_LEVEL_0</td>
<td>3</td>
<td>DB1_PDB1</td>
</tr>
<tr>
<td>300</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-25_3s1t4v1a</td>
<td>ORACOPYBKUPONFSXN_LEVEL_0</td>
<td>3</td>
<td>DB1_PDB1</td>
</tr>
<tr>
<td>305</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-26_451t4vt7</td>
<td>ORACOPYBKUPONFSXN_LEVEL_0</td>
<td>3</td>
<td>DB1_PDB1</td>
</tr>
<tr>
<td>299</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-27_461t4vt7</td>
<td>ORACOPYBKUPONFSXN_LEVEL_0</td>
<td>3</td>
<td>DB1_PDB1</td>
</tr>
<tr>
<td>310</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-28_471t4vt7</td>
<td>ORACOPYBKUPONFSXN_LEVEL_0</td>
<td>3</td>
<td>DB1_PDB1</td>
</tr>
<tr>
<td>303</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-29_481t4vt7</td>
<td>ORACOPYBKUPONFSXN_LEVEL_0</td>
<td>3</td>
<td>DB1_PDB1</td>
</tr>
<tr>
<td>309</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-30_491t5014</td>
<td>ORACOPYBKUPONFSXN_LEVEL_0</td>
<td>3</td>
<td>DB1_PDB1</td>
</tr>
<tr>
<td>301</td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-31_4a1t5015</td>
<td>ORACOPYBKUPONFSXN_LEVEL_0</td>
<td>3</td>
<td>DB1_PDB1</td>
</tr>
</tbody>
</table>
16. Switch database to copy to run recovery without database restore.

```sql
RMAN> switch database to copy;

Starting implicit crosscheck backup at 31-MAY-23
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=11 device type=DISK
Crosschecked 33 objects
Finished implicit crosscheck backup at 31-MAY-23

Starting implicit crosscheck copy at 31-MAY-23
using channel ORA_DISK_1
Crosschecked 68 objects
Finished implicit crosscheck copy at 31-MAY-23

searching for all files in the recovery area
cataloging files...
cataloging done

List of Cataloged Files
```
File Name:
/nfsfsxn/archlog/DB1/autobackup/2023_05_30/o1_mf_s_1138210401__08qlxrrr_.bkp

datafile 1 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-1_4f1t506m"
datafile 3 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-3_4g1t506m"
datafile 4 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-4_4h1t5083"
datafile 5 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-5_4q1t509n"
datafile 6 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-6_4m1t508t"
datafile 7 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOBS1_FNO-7_4u1t50a6"
datafile 8 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-8_4n1t509m"
datafile 9 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOBS1_FNO-9_4n1t509m"
datafile 10 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-10_4i1t5083"
datafile 11 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOBS1_FNO-11_4l1t508t"
datafile 12 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-12_4v1t50aa"
datafile 13 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-13_4o1t509m"
datafile 14 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-14_4j1t508s"
datafile 15 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOBS1_FNO-15_4r1t50a6"
datafile 16 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-16_501t50ad"
datafile 17 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-17_4p1t509m"
datafile 18 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-18_4k1t508t"
datafile 19 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOBS1_FNO-19_4s1t50a6"
datafile 20 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-20_511t50ad"
datafile 21 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-21_3o1t4ut2"
datafile 22 switched to datafile copy "/nfsfsxn/oracopy/data_D-
17. Run Oracle recovery up to last available archive log in flash recovery area.

```
RMAN> run {
  2> set until sequence=176;
  3> recover database;
  4> }
```

eexecuting command: SET until clause

Starting recover at 31-MAY-23
using channel ORA_DISK_1

starting media recovery

archived log for thread 1 with sequence 142 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_142__02n3x2qb_.ar
archived log for thread 1 with sequence 143 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_143__02rotwyb_.arc
archived log for thread 1 with sequence 144 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_144__02x563wh_.arc
archived log for thread 1 with sequence 145 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_145__031kg2co_.arc
archived log for thread 1 with sequence 146 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_146__035xpcdt_.arc
archived log for thread 1 with sequence 147 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_147__03bds8qf_.arc
archived log for thread 1 with sequence 148 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_148__03gyt7rx_.arc
archived log for thread 1 with sequence 149 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_149__03mfx17v_.arc
archived log for thread 1 with sequence 150 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_150__03qzz0ty_.arc
archived log for thread 1 with sequence 151 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_151__03wgxdry_.arc
archived log for thread 1 with sequence 152 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_152__040y85v3_.arc
archived log for thread 1 with sequence 153 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_153__04ox946w_.arc
archived log for thread 1 with sequence 154 is already on disk as
archived log for thread 1 with sequence 155 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_155__04tv1yvn_.arc
archived log for thread 1 with sequence 156 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_156__04xgfjtl_.arc
archived log for thread 1 with sequence 157 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_157__04zyg8hw_.arc
archived log for thread 1 with sequence 158 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_158__052gp9mt_.arc
archived log for thread 1 with sequence 159 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_159__0551wk7s_.arc
archived log for thread 1 with sequence 160 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_160__057146my_.arc
archived log for thread 1 with sequence 161 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_161__05b2dmwp_.arc
archived log for thread 1 with sequence 162 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_162__05drbj8n_.arc
archived log for thread 1 with sequence 163 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_163__05h8lmlh_.arc
archived log for thread 1 with sequence 164 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_164__05krsqmh_.arc
archived log for thread 1 with sequence 165 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_165__05n378pw_.arc
archived log for thread 1 with sequence 166 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_166__05pmg741_.ar

c
archived log for thread 1 with sequence 167 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_167__05s3o01r_.ar

c
archived log for thread 1 with sequence 168 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_168__05vmwt34_.ar

c
archived log for thread 1 with sequence 169 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_169__05y45qdd_.ar

c
archived log for thread 1 with sequence 170 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_170__060kg533_.ar

c
archived log for thread 1 with sequence 171 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_171__0631tvgv_.ar

c
archived log for thread 1 with sequence 172 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_172__065d94fq_.ar

c
archived log for thread 1 with sequence 173 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_173__067wnvy8_.ar

c
archived log for thread 1 with sequence 174 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_174__06b9zdh8_.ar

c
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_142__02n3x2qb_.ar
c
thread=1 sequence=142
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_143__02rotwy
b_.arc thread=1 sequence=143
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_144__02x563w
h_.arc thread=1 sequence=144
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_145__031kg2c
o_.arc thread=1 sequence=145
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_146__035xpcd
t_.arc thread=1 sequence=146
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_147__03bds8q
f_.arc thread=1 sequence=147
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_148__03gyt7r
x_.arc thread=1 sequence=148
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_149__03mfx17
v_.arc thread=1 sequence=149
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_150__03qzz0t
y_.arc thread=1 sequence=150
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_151__03wgxdr
y_.arc thread=1 sequence=151
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_152__040y85v
3_.arc thread=1 sequence=152
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_153__04ox946
w_.arc thread=1 sequence=153
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_154__04rbv7n
8_.arc thread=1 sequence=154
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_155__04tv1yv
n_.arc thread=1 sequence=155
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_156__04xgfjt
l_.arc thread=1 sequence=156
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_157__04zyg8h
w_.arc thread=1 sequence=157
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_158__052gp9m
t_.arc thread=1 sequence=158
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_159__0551wk7s_.arc thread=1 sequence=159
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_160__057146my_.arc thread=1 sequence=160
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_161__05b2dmwp_.arc thread=1 sequence=161
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_162__05drbj8n_.arc thread=1 sequence=162
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_163__05h81mlm_.arc thread=1 sequence=163
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_164__05krsqmh_.arc thread=1 sequence=164
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_165__05n378pw_.arc thread=1 sequence=165
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_166__05pmg74l_.arc thread=1 sequence=166
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_167__05s3001r_.arc thread=1 sequence=167
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_168__05vmwt34_.arc thread=1 sequence=168
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_169__05y45qd_d_.arc thread=1 sequence=169
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_170__060kg33_.arc thread=1 sequence=170
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_171__0631tvgv_.arc thread=1 sequence=171
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_172__065d94fq_.arc thread=1 sequence=172
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_173__067wnwy8_.arc thread=1 sequence=173
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_174__06b9zd8_.arc thread=1 sequence=174
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_175__08c7jc2b_.arc thread=1 sequence=175
media recovery complete, elapsed time: 00:48:34
Finished recover at 31-MAY-23

For faster recovery, enable parallel sessions with recovery_parallelism parameter or specify degree of parallel in recovery command for database recovery: RECOVER DATABASE PARALLEL (DEGREE d INSTANCES DEFAULT); In general, degrees of parallelism should be equal to number of CPU cores on the host.

18. Exit RMAN, login to Oracle as oracle user via sqlplus to open database and reset log after an incomplete recovery.
SQL> select name, open_mode from v$database;

NAME      OPEN_MODE
--------- --------------------
DB1       MOUNTED

SQL> select member from v$logfile;

MEMBER
--------------------------------------------------------------------
------------
+DATA/DB1/ONLINELOG/group_3.264.1136666437
+DATA/DB1/ONLINELOG/group_2.263.1136666437
+DATA/DB1/ONLINELOG/group_1.262.1136666437

SQL> alter database rename file '+DATA/DB1/ONLINELOG/group_1.262.1136666437' to '/nfsfsxn/oracopy/redo01.log';
Database altered.

SQL> alter database rename file '+DATA/DB1/ONLINELOG/group_2.263.1136666437' to '/nfsfsxn/oracopy/redo02.log';
Database altered.

SQL> alter database rename file '+DATA/DB1/ONLINELOG/group_3.264.1136666437' to '/nfsfsxn/oracopy/redo03.log';
Database altered.

SQL> alter database open resetlogs;
Database altered.

19. Validate the database restored to new host that has the row we have inserted before primary database failure.
SQL> show pdb

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>CON_NAME</th>
<th>OPEN MODE</th>
<th>RESTRICTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>READ ONLY</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>DB1_PDB1</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>4</td>
<td>DB1_PDB2</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>5</td>
<td>DB1_PDB3</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
</tbody>
</table>

SQL> alter session set container=db1_pdb1;
Session altered.

SQL> select * from test;

<table>
<thead>
<tr>
<th>ID</th>
<th>DT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>18-MAY-23 02.35.37.000000 PM</td>
</tr>
<tr>
<td>2</td>
<td>30-MAY-23 05.23.11.000000 PM</td>
</tr>
</tbody>
</table>

---

test oracle incremental merge switch to copy
test recovery on a new EC2 instance host with image copy on FSxN

20. Other post recovery tasks

Add FSxN NFS mount to fstab so that the NFS file system will be mounted when EC2 instance host rebooted.

As EC2 user, vi /etc/fstab and add following entry:

```
172.30.15.19:/ora_01_copy /nfsfsxn nfs rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr 0 0
```

Update the Oracle init file from primary database init file backup that is restored to /tmp/archive and create spfile as needed.

This completes the Oracle VLDB database recovery from backup image copy on FSxN NFS file system to a new EC2 DB instance host.
Clone Oracle standby image copy for other use cases
Another benefit of using AWS FSx ONTAP for staging Oracle VLDB image copy is that it can be FlexCloned to serve many other purposes with minimal additional storage investment. In the following use case, we demonstrate how to snapshot and clone the staging NFS volume on FSx ONTAP for other Oracle use cases such as DEV, UAT, etc.

1. We begin with inserting a row into the same test table we have created before.

   SQL> insert into test values (3, sysdate, 'test clone on a new EC2 instance host with image copy on FSxN');
   1 row created.

   SQL> select * from test;
   ID
   ----------
   DT
   -------------------------------
   -------
   EVENT
   -------------------------------
   -------
   1
   18-MAY-23 02.35.37.000000 PM
test oracle incremental merge switch to copy

   2
   30-MAY-23 05.23.11.000000 PM
test recovery on a new EC2 instance host with image copy on FSxN

   ID
   ----------
   DT
   -------------------------------
   -------
   EVENT
   -------------------------------
   -------
   3
   05-JUN-23 03.19.46.000000 PM
test clone on a new EC2 instance host with image copy on FSxN

   SQL>
2. Take a RMAN backup and merge to FSx ONTAP database image copy so that the transaction will be captured in the backup set on FSx NFS mount but not merged into copy until cloned database is recovered.

RMAN> @/home/oracle/rman_bkup_merge.cmd

3. Login to FSx cluster via ssh as fsxadmin user to observe the snapshots created by scheduled backup policy - oracle and take an one-off snapshot so that it will include the transaction we committed in step 1.
FtxId06c3c8b2a7bd56458::> vol snapshot create -vserver svm_ora
-volume ora_01_copy -snapshot one-off.2023-06-05-1137 -foreground true

FtxId06c3c8b2a7bd56458::> snapshot show

---Blocks---

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Volume</th>
<th>Snapshot</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>---------</td>
<td>----------</td>
<td>----------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>svm_ora</td>
<td>ora_01_copy</td>
<td>daily.2023-06-02_0010</td>
<td>3.59GB</td>
</tr>
<tr>
<td>2%</td>
<td>5%</td>
<td>daily.2023-06-03_0010</td>
<td>1.10GB</td>
</tr>
<tr>
<td>1%</td>
<td>1%</td>
<td>daily.2023-06-04_0010</td>
<td>608KB</td>
</tr>
<tr>
<td>0%</td>
<td>0%</td>
<td>daily.2023-06-05_0010</td>
<td>3.81GB</td>
</tr>
<tr>
<td>2%</td>
<td>5%</td>
<td>one-off.2023-06-05-1137</td>
<td>168KB</td>
</tr>
<tr>
<td>0%</td>
<td>0%</td>
<td>svm_ora_root</td>
<td></td>
</tr>
<tr>
<td>0%</td>
<td>78%</td>
<td>weekly.2023-05-28_0015</td>
<td>1.86MB</td>
</tr>
<tr>
<td>0%</td>
<td>22%</td>
<td>daily.2023-06-04_0010</td>
<td>152KB</td>
</tr>
<tr>
<td>0%</td>
<td>70%</td>
<td>weekly.2023-06-04_0015</td>
<td>1.24MB</td>
</tr>
<tr>
<td>0%</td>
<td>27%</td>
<td>daily.2023-06-05_0010</td>
<td>196KB</td>
</tr>
<tr>
<td>0%</td>
<td>22%</td>
<td>hourly.2023-06-05_1005</td>
<td>156KB</td>
</tr>
<tr>
<td>0%</td>
<td>22%</td>
<td>hourly.2023-06-05_1105</td>
<td>156KB</td>
</tr>
<tr>
<td>0%</td>
<td>22%</td>
<td>hourly.2023-06-05_1205</td>
<td>156KB</td>
</tr>
<tr>
<td>0%</td>
<td>22%</td>
<td>hourly.2023-06-05_1305</td>
<td>156KB</td>
</tr>
<tr>
<td>0%</td>
<td>22%</td>
<td>hourly.2023-06-05_1405</td>
<td>1.87MB</td>
</tr>
<tr>
<td>0%</td>
<td>78%</td>
<td>hourly.2023-06-05_1505</td>
<td>148KB</td>
</tr>
<tr>
<td>0%</td>
<td>22%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>15 entries were displayed.</td>
<td></td>
</tr>
</tbody>
</table>

4. Clone from the one-off snapshot to be used for standing up a new DB1 clone instance on an alternative EC2 Oracle host. You have the option to clone from any available daily snapshots for volume ora_01_copy.

```bash
FsxId06c3c8b2a7bd56458::> vol clone create -flexclone db1_20230605of -type RW -parent-vserver svm_ora -parent-volume ora_01_copy -junction-path /db1_20230605of -junction-active true -snapshot one-off.2023-06-05-1137
[Job 464] Job succeeded: Successful

FsxId06c3c8b2a7bd56458::>
FsxId06c3c8b2a7bd56458::> vol show db1*
Vserver   Volume       Aggregate    State      Type       Size    Available Used%  
----------- ------------- ----------- --------- ---- ----------
----------- -----        -------- --------- ------- -------
svm_ora   db1_20230605of
          aggr1        online     RW        200GB 116.6GB  38%
FsxId06c3c8b2a7bd56458::>
```

5. Turn off snapshot policy for the cloned volume as it inherits parent volume snapshot policy unless you want to protect the cloned volume, then leave it alone.

```bash
FsxId06c3c8b2a7bd56458::> vol modify -volume db1_20230605of -snapshot-policy none
Warning: You are changing the Snapshot policy on volume "db1_20230605of" to "none". Snapshot copies on this volume that do not match any of the prefixes of the new Snapshot policy will not be deleted. However, when the new Snapshot policy takes effect, depending on the new retention count, any existing Snapshot copies that continue to use the same prefixes might be deleted. See the 'volume modify' man page for more information.
Do you want to continue? {y|n}: y
Volume modify successful on volume db1_20230605of of Vserver svm_ora.

FsxId06c3c8b2a7bd56458::>
```

6. Login to a new EC2 Linux instance with Oracle software pre-installed with same version and patch level as your primary Oracle EC2 instance and mount the cloned volume.
7. Validate the database incremental backup sets, image copy, and available archived logs on FSx NFS mount.

```
[ec2-user@ip-172-30-15-124 ~]$ ls -ltr /nfsfsxn/oracopy
```

```
total 79450332
-rw-r----- 1 oracle 54331 482353152 Jun  1 19:02 data_D-DB1_I-1730530050_TS-SYSAUX_FNO-6_891tkrhr
-rw-r----- 1 oracle 54331 419438592 Jun  1 19:03 data_D-DB1_I-1730530050_TS-SYSTEM_FNO-5_8d1tkril
-rw-r----- 1 oracle 54331 241180672 Jun  1 19:03 data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-8_8g1tkrj7
-rw-r----- 1 oracle 54331 912506880 Jun  1 20:21 8n1tkvv2_279_1_1
-rw-r----- 1 oracle 54331 925696 Jun  1 20:21 8q1tl05i_282_1_1
-rw-r----- 1 oracle 54331 1169014784 Jun  1 20:21 8p1tkvv2_281_1_1
-rw-r----- 1 oracle 54331 6455296 Jun  1 20:21 8r1tl05m_283_1_1
-rw-r----- 1 oracle 54331 139264 Jun  1 20:21 8t1tl05t_284_1_1
-rw-r----- 1 oracle 54331 3514368 Jun  1 20:21 8s1tl05t_285_1_1
-rw-r----- 1 oracle 54331 139264 Jun  1 20:21 8u1tl060_286_1_1
-rw-r----- 1 oracle 54331 425984 Jun  1 20:21 8q1tl105_287_1_1
-rw-r----- 1 oracle 54331 344064 Jun  1 20:21 8r1tl105_289_1_1
-rw-r----- 1 oracle 54331 245760 Jun  1 20:21 9s1tl05t_290_1_1
-rw-r----- 1 oracle 54331 237568 Jun  1 20:21 9t1tl05t_291_1_1
-rw-r----- 1 oracle 54331 57344 Jun  1 20:21 911tl066_292_1_1
-rw-r----- 1 oracle 54331 534781952 Jun  1 20:21 921tl065_294_1_1
-rw-r----- 1 oracle 54331 534781952 Jun  1 20:21 931tl063_295_1_1
-rw-r----- 1 oracle 54331 534781952 Jun  1 20:21 941tl064_296_1_1
-rw-r----- 1 oracle 54331 1040760832 Jun  1 20:23 8n1tkvv2_278_1_1
-rw-r----- 1 oracle 54331 932847616 Jun  1 20:24 8o1tkvv2_280_1_1
-rw-r----- 1 oracle 54331 1121984512 Jun  1 15:21 data_D-DB1_I-1730530050_TS-SYSTEM_FNO-1_821tkrb8
-rw-r----- 1 oracle 54331 1027612672 Jun  1 15:21 data_D-DB1_I-1730530050_TS-SYSAUX_FNO-3_831tkrd9
-rw-r----- 1 oracle 54331 429924352 Jun  1 15:21 data_D-DB1_I-1730530050_TS-SYSTEM_FNO-9_8a1tkrhr
-rw-r----- 1 oracle 54331 707796992 Jun  1 15:21 data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-4_851tkrgf
-rw-r----- 1 oracle 54331 534781952 Jun  1 15:21 data_D-DB1_I-1730530050_TS-SYSAUX_FNO-14_871tkrhr
-rw-r----- 1 oracle 54331 534781952 Jun  1 15:21 data_D-DB1_I-1730530050_TS-SYSAUX_FNO-18_881tkrhr
```
8. The recovery processes now are similar to previous use case of recovery to a new EC2 DB instance after a failure - set oracle environment (oratab, $ORACLE_HOME, $ORACLE_SID) to match with primary production instance, create an init file including db_recovery_file_dest_size and db_recovery_file_dest that point to flash recovery directory on FSx NFS mount. Then, launch RMAN to run recovery. Following are command steps and output.

```
[oracle@ip-172-30-15-124 dbs]$ rman target / nocatalog

Recovery Manager: Release 19.0.0.0.0 - Production on Wed Jun 7 14:44:33 2023
Version 19.18.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.

connected to target database (not started)

RMAN> startup nomount;

Oracle instance started

Total System Global Area   10737418000 bytes

Fixed Size               9174800 bytes
Variable Size            1577058304 bytes
Database Buffers         9126805504 bytes
Redo Buffers             24379392 bytes

RMAN> set dbid = 1730530050;

executing command: SET DBID

RMAN> restore controlfile from autobackup;

Starting restore at 07-JUN-23
allocated channel: ORA_DISK_1
```
channel ORA_DISK_1: SID=2 device type=DISK

recovery area destination: /nfsfsxn/archlog/
database name (or database unique name) used for search: DB1

channel ORA_DISK_1: AUTOBACKUP
/nfsfsxn/archlog/DB1/autobackup/2023_06_05/o1_mf_s_1138721482_l7vzybvq_.bkp found in the recovery area
channel ORA_DISK_1: looking for AUTOBACKUP on day: 20230607
channel ORA_DISK_1: looking for AUTOBACKUP on day: 20230606
channel ORA_DISK_1: looking for AUTOBACKUP on day: 20230605
channel ORA_DISK_1: restoring control file from AUTOBACKUP
/nfsfsxn/archlog/DB1/autobackup/2023_06_05/o1_mf_s_1138721482_l7vzybvq_.bkp
channel ORA_DISK_1: control file restore from AUTOBACKUP complete
output file name=/nfsfsxn/oracopy/db1.ctl
Finished restore at 07-JUN-23

RMAN> alter database mount;
released channel: ORA_DISK_1
Statement processed

RMAN> list incarnation;

List of Database Incarnations
DB Key Inc Key DB Name DB ID STATUS Reset SCN Reset Time
------- ------- -------- ---------------- --- ---------- ----------
1       1       DB1      1730530050       PARENT 1          17-APR-19
2       2       DB1      1730530050       CURRENT 1920977    12-MAY-23

RMAN> list copy of database tag 'OraCopyBKUPonFSxN_level_0';

List of Datafile Copies
=======================
Key   File S Completion Time Ckp SCN    Ckp Time        Sparse
------- ----- --------------- ---------- --------------- -----
362     1    A 05-JUN-23       8319160    01-JUN-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSTEM_FNO-1_821tkrb8
   Tag: ORACOPYBKUPONFSXN_LEVEL_0
363     3    A 05-JUN-23       8319165    01-JUN-23       NO
365  4  A  05-JUN-23       8319171    01-JUN-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSAUX_FNO-3_831tkrd9
   Tag: ORACOPYBKUPONFSXN_LEVEL_0

355  5  A  01-JUN-23       2383520    12-MAY-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-4_851tkrgf
   Tag: ORACOPYBKUPONFSXN_LEVEL_0

349  6  A  01-JUN-23       2383520    12-MAY-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSAUX_FNO-6_891tkrhr
   Tag: ORACOPYBKUPONFSXN_LEVEL_0
   Container ID: 2, PDB Name: PDB$SEED

372  7  A  05-JUN-23       8319201    01-JUN-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
7_8h1tkrj9
   Tag: ORACOPYBKUPONFSXN_LEVEL_0

361  8  A  01-JUN-23       2383520    12-MAY-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-8_8gltkrj7
   Tag: ORACOPYBKUPONFSXN_LEVEL_0
   Container ID: 2, PDB Name: PDB$SEED

364  9  A  05-JUN-23       8318717    01-JUN-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSTEM_FNO-9_8a1tkhr
   Tag: ORACOPYBKUPONFSXN_LEVEL_0
   Container ID: 3, PDB Name: DB1_PDB1

376 10  A  05-JUN-23       8318714    01-JUN-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSAUX_FNO-10_861tkrgo
   Tag: ORACOPYBKUPONFSXN_LEVEL_0
   Container ID: 3, PDB Name: DB1_PDB1

377 11  A  05-JUN-23       8318720    01-JUN-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSTEM_FNO-11_891tkrhr
   Tag: ORACOPYBKUPONFSXN_LEVEL_0
   Container ID: 3, PDB Name: DB1_PDB1

UNDOTBS1_FNO-11_841tkrf2
  Tag: ORACOPYBKUPONFSXN_LEVEL_0
  Container ID: 3, PDB Name: DB1_PDB1
375 12 A 05-JUN-23 8318719 01-JUN-23 NO
  Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
  12_8i1tkrj9
  Tag: ORACOPYBKUPONFSXN_LEVEL_0
  Container ID: 3, PDB Name: DB1_PDB1
368 13 A 05-JUN-23 8319184 01-JUN-23 NO
  Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
  SYSTEM_FNO-13_8b1tkril
  Tag: ORACOPYBKUPONFSXN_LEVEL_0
  Container ID: 4, PDB Name: DB1_PDB2
366 14 A 05-JUN-23 8319175 01-JUN-23 NO
  Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
  SYSAUX_FNO-14_871tkrhr
  Tag: ORACOPYBKUPONFSXN_LEVEL_0
  Container ID: 4, PDB Name: DB1_PDB2
370 15 A 05-JUN-23 8319193 01-JUN-23 NO
  Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
  UNDOTBS1_FNO-15_8eltkril
  Tag: ORACOPYBKUPONFSXN_LEVEL_0
  Container ID: 4, PDB Name: DB1_PDB2
373 16 A 05-JUN-23 8319206 01-JUN-23 NO
  Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
  16_8j1tkrja
  Tag: ORACOPYBKUPONFSXN_LEVEL_0
  Container ID: 4, PDB Name: DB1_PDB2
369 17 A 05-JUN-23 8319188 01-JUN-23 NO
  Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
  SYSTEM_FNO-17_8c1tkril
  Tag: ORACOPYBKUPONFSXN_LEVEL_0
  Container ID: 5, PDB Name: DB1_PDB3
367 18 A 05-JUN-23 8319180 01-JUN-23 NO
  Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
  SYSAUX_FNO-18_881tkrhr
  Tag: ORACOPYBKUPONFSXN_LEVEL_0
  Container ID: 5, PDB Name: DB1_PDB3
371  19  A 05-JUN-23  8319197  01-JUN-23  NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-19_8f1tkrj4
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 5, PDB Name: DB1_PDB3

374  20  A 05-JUN-23  8319210  01-JUN-23  NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-20_8k1tkrjb
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 5, PDB Name: DB1_PDB3

378  21  A 05-JUN-23  8318720  01-JUN-23  NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-21_7j1tkqk6
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

388  22  A 05-JUN-23  8318714  01-JUN-23  NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-22_7k1tkqk6
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

384  23  A 05-JUN-23  8318717  01-JUN-23  NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-23_7l1tkqk6
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

389  24  A 05-JUN-23  8318719  01-JUN-23  NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-24_7m1tkqk6
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

381  25  A 05-JUN-23  8318720  01-JUN-23  NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-25_7n1tkqrh
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

392  26  A 05-JUN-23  8318714  01-JUN-23  NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-26_7o1tkqrj
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

385  27   A  05-JUN-23  8318717  01-JUN-23  NO
    Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
          27_7p1tkqrq
    Tag: ORACOPYBKUPONFSXN_LEVEL_0
    Container ID: 3, PDB Name: DB1_PDB1

390  28   A  05-JUN-23  8318719  01-JUN-23  NO
    Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
          28_7q1tkqs1
    Tag: ORACOPYBKUPONFSXN_LEVEL_0
    Container ID: 3, PDB Name: DB1_PDB1

380  29   A  05-JUN-23  8318720  01-JUN-23  NO
    Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
          29_7r1tkr32
    Tag: ORACOPYBKUPONFSXN_LEVEL_0
    Container ID: 3, PDB Name: DB1_PDB1

391  30   A  05-JUN-23  8318714  01-JUN-23  NO
    Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
          30_7s1tkr3a
    Tag: ORACOPYBKUPONFSXN_LEVEL_0
    Container ID: 3, PDB Name: DB1_PDB1

382  31   A  05-JUN-23  8318717  01-JUN-23  NO
    Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
          31_7t1tkr3i
    Tag: ORACOPYBKUPONFSXN_LEVEL_0
    Container ID: 3, PDB Name: DB1_PDB1

387  32   A  05-JUN-23  8318719  01-JUN-23  NO
    Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
          32_7u1tkr42
    Tag: ORACOPYBKUPONFSXN_LEVEL_0
    Container ID: 3, PDB Name: DB1_PDB1

383  33   A  05-JUN-23  8318719  01-JUN-23  NO
    Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
          33_7v1tkra6
    Tag: ORACOPYBKUPONFSXN_LEVEL_0
    Container ID: 3, PDB Name: DB1_PDB1

379  34   A  05-JUN-23  8318717  01-JUN-23  NO
    Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

386  35   A 05-JUN-23       8318714    01-JUN-23       NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-

Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

RMAN> switch database to copy;

datafile 1 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-1_821tkrb8"
datafile 3 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-3_831tkrd9"
datafile 4 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-4_851tkrgf"
datafile 5 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-5_8d1tkril"
datafile 6 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-6_891tkrhr"
datafile 7 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-USERS_FNO-7_8h1tkrj9"
datafile 8 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-8_8g1tkrj7"
datafile 9 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-9_8a1tkrhr"
datafile 10 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SYSAUX_FNO-10_861tkrgo"
datafile 11 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-UNDOTBS1_FNO-11_841tkrf2"
datafile 12 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-USERS_FNO-12_8i1tkrj9"
datafile 13 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SYSTEM_FNO-13_8b1tkril"
datafile 14 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SYSAUX_FNO-14_871tkhrhr"
datafile 15 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-UNDOTBS1_FNO-15_8e1tkril"
datafile 16 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-USERS_FNO-16_8j1tkrja"
datafile 17 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SYSTEM_FNO-17_8cltkril"
datafile 18 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SYSAUX_FNO-18_881tkrhr"
RMAN> run {
  2> set until sequence 204;
  3> recover database;
  4> }

executing command: SET until clause

Starting recover at 07-JUN-23
using channel ORA_DISK_1

starting media recovery
archived log for thread 1 with sequence 190 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_190_l7vwvvt9_.arc
archived log for thread 1 with sequence 191 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_191_l7vx6vmg_.arc
archived log for thread 1 with sequence 192 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_192_l7vxctms_.arc
archived log for thread 1 with sequence 193 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_193_l7vxjjps_.arc
archived log for thread 1 with sequence 194 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_194_l7vxnxrh_.arc
archived log for thread 1 with sequence 195 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_195_l7vxswv5_.arc
archived log for thread 1 with sequence 196 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_196_l7vxy1wp_.arc
archived log for thread 1 with sequence 197 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_197_l7vy3cyw_.arc
archived log for thread 1 with sequence 198 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_198_l7vy8245_.arc
archived log for thread 1 with sequence 199 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_199_l7vydv4c_.arc
archived log for thread 1 with sequence 200 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_200_l7vykf23_.arc
archived log for thread 1 with sequence 201 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_201_l7vyp1dh_.arc
archived log for thread 1 with sequence 202 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_202_l7vyrvm5_.arc
archived log for thread 1 with sequence 203 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_203_l7vzdfrm_.arc
archived log file name=/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_190_l7vwvvt9
_.arc thread=1 sequence=190
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_191_l7vx6vmg_.arc thread=1 sequence=191 archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_192_l7vxctms_.arc thread=1 sequence=192 archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_193_l7vxjjps_.arc thread=1 sequence=193 archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_194_l7vxnxrh_.arc thread=1 sequence=194 archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_195_l7vxswv5_.arc thread=1 sequence=195 archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_196_l7vy1lwp_.arc thread=1 sequence=196 archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_197_l7vy3cyw_.arc thread=1 sequence=197 archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_198_l7vy8245_.arc thread=1 sequence=198 archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_199_l7vydv4c_.arc thread=1 sequence=199 archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_200_l7ykf23_.arc thread=1 sequence=200 archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_201_l7vyp1dh_.arc thread=1 sequence=201 archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_202_l7vym5_.arc thread=1 sequence=202 archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_203_l7zdffwm_.arc thread=1 sequence=203 archived log file
media recovery complete, elapsed time: 00:19:30
Finished recover at 07-JUN-23

RMAN> exit

Recovery Manager complete.
[oracle@ip-172-30-15-124 dbs]$ sqlplus / as sysdba
SQL> select member from v$logfile;

MEMBER
---------------------------------------------------------------
+DATA/DB1/ONLINELOG/group_3.264.1136666437
+DATA/DB1/ONLINELOG/group_2.263.1136666437
+DATA/DB1/ONLINELOG/group_1.262.1136666437

SQL> alter database rename file '+DATA/DB1/ONLINELOG/group_1.262.1136666437' to '/nfsfsxn/oracopy/redo01.log';
Database altered.

SQL> alter database rename file '+DATA/DB1/ONLINELOG/group_2.263.1136666437' to '/nfsfsxn/oracopy/redo02.log';
Database altered.

SQL> alter database rename file '+DATA/DB1/ONLINELOG/group_3.264.1136666437' to '/nfsfsxn/oracopy/redo03.log';
Database altered.

SQL> alter database noarchivelog;
Database altered.

SQL> alter database open resetlogs;
Database altered.

SQL> set lin 200;
SQL> select name from v$datafile
  2  union
  3  select name from v$controlfile
  4  union
  5  select name from v$tempfile
  6  union
  7  select member from v$logfile;

NAME
--------------------------------------------
/nfsfsxn/oracopy/DB1/FB864A929AEB79B9E053630F1EAC7046/datafile/o1_mf_temp_181bhz6g_.tmp
/nfsfsxn/oracopy/DB1/FB867DA8C68C816EE053630F1EAC2BCF/datafile/o1_mf_temp_181bj16t_.tmp
/nfsfsxn/oracopy/DB1/FB867EA89ECF81C0E053630F1EACB901/datafile/o1_mf_temp_181bj135_.tmp
/nfsfsxn/oracopy/DB1/FB867F8A4D4F821CE053630F1EAC69CC/datafile/o1_mf_temp_181bj13g_.tmp
/nfsfsxn/oracopy/DB1/datafile/o1_mf_temp_181bhwjg_.tmp
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-21_7j1tkqk6
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-22_7k1tkqk6
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-23_7l1tkqk6
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-24_7m1tkqk6
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-25_7n1tkqrp
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-26_7o1tkqrj
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-27_7p1tkqrq
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-28_7q1tkqs1
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-29_7r1tkr32
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-30_7s1tkr3a
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-31_7t1tkr3i
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-32_7u1tkr42
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-33_7v1tkra6
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-34_801tkram
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-35_811tkrap
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-10_861tkrgeo
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-14_871tkrhr
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-18_881tkrhr
SQL> show pdbs;

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>CON_NAME</th>
<th>OPEN MODE</th>
<th>RESTRICTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>READ ONLY</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>DB1_PDB1</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>4</td>
<td>DB1_PDB2</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>5</td>
<td>DB1_PDB3</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
</tbody>
</table>

SQL> alter session set container=db1_pdb1;

Session altered.

SQL> select * from test;

<table>
<thead>
<tr>
<th>ID</th>
<th>DT</th>
<th>EVENT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
9. Rename the cloned database instance and change database ID with Oracle nid utility. The database instance state needs to be in `mount` to execute the command.

```sql
SQL> select name, open_mode, log_mode from v$database;

NAME      OPEN_MODE            LOG_MODE
--------- -------------------- ------------
DB1       READ WRITE           NOARCHIVELOG

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> startup mount;
ORACLE instance started.

Total System Global Area 1.0737E+10 bytes
Fixed Size                  9174800 bytes
Variable Size            1577058304 bytes
Database Buffers         9126805504 bytes
Redo Buffers               24379392 bytes
Database mounted.
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.18.0.0.0
[oracle@ip-172-30-15-124 dbs]$ nid target=/ dbname=db1tst

DBNEWID: Release 19.0.0.0.0 - Production on Wed Jun 7 16:15:14 2023
Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.

Connected to database DB1 (DBID=1730530050)
```
Connected to server version 19.18.0

Control Files in database:
    /nfsfsxn/oracopy/db1.ctl

Change database ID and database name DB1 to DB1TST? (Y/[N]) => Y

Proceeding with operation
Changing database ID from 1730530050 to 3054879890
Changing database name from DB1 to DB1TST
    Control File /nfsfsxn/oracopy/db1.ctl - modified
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-
        1_821tkrb - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-
        3_831tkrd - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
        UNDOTBS1_FNO-4_851tkrg - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-
        5_8d1tkri - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-
        6_891tkrh - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDO_TBLF File
        7_8h1tkrj - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-
        9_8a1tkri - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-
        10_881tkrg - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
        UNDOTBS1_FNO-11_8a1tkri - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-
        12_8g1tkrg - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-
        13_8h1tkri - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
        UNDOTBS1_FNO-14_881tkrg - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-
        15_8j1tkrg - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-
        16_881tkri - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
        UNDOTBS1_FNO-17_8g1tkri - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-
        18_8h1tkrg - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-20_8k1tkrj - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-21_7j1tkqk - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-22_7k1tkqk - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-23_7l1tkqk - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-24_7m1tkqk - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-25_7n1tkqr - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-26_7o1tkqr - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-27_7p1tkqr - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-28_7q1tkqs - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-29_7r1tkr3 - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-30_7s1tkr3 - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-31_7t1tkr3 - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-32_7u1tkr4 - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-33_7v1tkra - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-34_801tkra - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-35_811tkra - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/DB1/datafile/o1_mf_temp_181bhwjg_.tm - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/DB1/FB864A929AEB79B9E053630F1EAC7046/datafile/o1_mf_temp_181bh26g_.tm - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/DB1/FB867DA8C68C816EE053630F1EAC2BCF/datafile/o1_mf_temp_181bj16t_.tm - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/DB1/FB867EA89ECF81C0E053630F1EACB901/datafile/o1_mf_temp_181bj135_.tm - dbid changed, wrote new name
Datafile /nfsfsxn/oracopy/DB1/FB867F8A4D4F821CE053630F1EAC69CC/datafile/o1_mf
Database name changed to DB1TST.
Modify parameter file and generate a new password file before restarting.
Database ID for database DB1TST changed to 3054879890.
All previous backups and archived redo logs for this database are unusable.
Database is not aware of previous backups and archived logs in Recovery Area.
Database has been shutdown, open database with RESETLOGS option.
Successfully changed database name and ID.
DBNEWID - Completed successfully.

10. Change Oracle database environment configuration to new database name or instance ID in oratab, init file, and create necessary admin directories that match with new instance ID. Then, start the instance with resetlogs option.
SQL> startup mount;
ORACLE instance started.

Total System Global Area 1.0737E+10 bytes
Fixed Size 9174800 bytes
Variable Size 1577058304 bytes
Database Buffers 9126805504 bytes
Redo Buffers 24379392 bytes
Database mounted.
SQL> alter database open resetlogs;
Database altered.

SQL> select name, open_mode, log_mode from v$database;

<table>
<thead>
<tr>
<th>NAME</th>
<th>OPEN_MODE</th>
<th>LOG_MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB1TST</td>
<td>READ WRITE</td>
<td>NOARCHIVELOG</td>
</tr>
</tbody>
</table>

SQL> show pdbs

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>CON_NAME</th>
<th>OPEN_MODE</th>
<th>RESTRICTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>READ ONLY</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>DB1_PDB1</td>
<td>MOUNTED</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>DB1_PDB2</td>
<td>MOUNTED</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>DB1_PDB3</td>
<td>MOUNTED</td>
<td></td>
</tr>
</tbody>
</table>

SQL> alter pluggable database all open;
Pluggable database altered.

SQL> show pdbs

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>CON_NAME</th>
<th>OPEN_MODE</th>
<th>RESTRICTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>READ ONLY</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>DB1_PDB1</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>4</td>
<td>DB1_PDB2</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>5</td>
<td>DB1_PDB3</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
</tbody>
</table>

This completes the clone of a new Oracle instance from staging database copy on FSx NFS mount for DEV, UAT, or any other use cases. Multiple Oracle instances can be cloned off the same staging image copy.
If you run into error RMAN-06571: datafile 1 does not have recoverable copy when switching the database to copy, check database incarnation that matches with primary production DB. If needed, reset the incarnation to match with primary with RMAN command reset database to incarnation n;.

Where to find additional information

To learn more about the information described in this document, review the following documents and/or websites:

- RMAN: Merged Incremental Backup Strategies (Doc ID 745798.1)
  https://support.oracle.com/knowledge/Oracle%20Database%20Products/745798_1.html

- RMAN Backup and Recovery User’s Guide

- Amazon FSx for NetApp ONTAP
  https://aws.amazon.com/fsx/netapp-ontap/

- Amazon EC2
  https://aws.amazon.com/pm/ec2/?trk=36c6da98-7b20-48fa-8225-4784bcded9843&sc_channel=ps&kwcid=AL14422!3!467723097970le!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfbhCKARIsAJzSr0qwhn6l71jiWzSeaT9Uhlv-Y-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:G:s&kwcid=AL14422!3!467723097970le!!g!!aws%20ec2

TR-4974: Oracle 19c in Standalone Restart on AWS FSx/EC2 with NFS/ASM

Allen Cao, Niyaz Mohamed, NetApp

Purpose

ASM (Automatic Storage Management) is a popular Oracle storage volume manager that is employed in many Oracle installations. It is also Oracle’s recommended storage management solution. It provides an alternative to conventional volume managers and file systems. Since Oracle version 11g, ASM has been packaged with grid infrastructure rather than a database. As a result, in order to utilize Oracle ASM for storage management without RAC, you must install Oracle grid infrastructure in a standalone server, also known as Oracle Restart. Doing so certainly adds more complexity in an otherwise simpler Oracle database deployment. However, as the name implies, when Oracle is deployed in Restart mode, any failed Oracle services are restarted after a host reboot without user intervention, which provides a certain degree of high availability or HA functionality.

Oracle ASM is generally deployed in FC, iSCSI storage protocols and luns as raw storage devices. However, ASM on NFS protocol and NFS file system is also supported configuration by Oracle. In this documentation, we demonstrate how to deploy an Oracle 19c database with the NFS protocol and Oracle ASM in an Amazon FSx for ONTAP storage environment with EC2 compute instances. We also demonstrate how to use the NetApp SnapCenter service through the NetApp BlueXP console to backup, restore, and clone your Oracle database for dev/test or other use cases for storage-efficient database operation in the AWS public cloud.

This solution addresses the following use cases:
- Oracle database deployment in Amazon FSx for ONTAP storage and EC2 compute instances with NFS/ASM
- Testing and validating an Oracle workload in the public AWS cloud with NFS/ASM
- Testing and validating Oracle database Restart functionalities deployed in AWS

**Audience**

This solution is intended for the following people:

- A DBA who would like to deploy Oracle in an AWS public cloud with NFS/ASM.
- A database solution architect who would like to test Oracle workloads in the AWS public cloud.
- The storage administrator who would like to deploy and manage an Oracle database deployed to AWS FSx storage.
- The application owner who would like to stand up an Oracle database in AWS FSx/EC2.

**Solution test and validation environment**

The testing and validation of this solution was performed in an AWS FSx and EC2 environment that might not match the final deployment environment. For more information, see the section [Key Factors for Deployment Consideration].

**Architecture**

![Oracle Deployment in AWS FSx/EC2 with NFS/ASM](image)

**Hardware and software components**

<table>
<thead>
<tr>
<th>Hardware</th>
<th>FSx ONTAP storage</th>
<th>Current version offered by AWS</th>
<th>One FSx HA cluster in the same VPC and availability zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSx ONTAP storage</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
EC2 instance for compute | t2.xlarge/4vCPU/16G | Two EC2 T2 xlarge EC2 instances, one as primary DB server and the other as a clone DB server

**Software**

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RedHat Linux</td>
<td>RHEL-8.6.0_HVM-20220503-x86_64-2-Hourly2-GP2</td>
<td>Deployed RedHat subscription for testing</td>
</tr>
<tr>
<td>Oracle Grid Infrastructure</td>
<td>Version 19.18</td>
<td>Applied RU patch p34762026_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle Database</td>
<td>Version 19.18</td>
<td>Applied RU patch p34765931_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle OPatch</td>
<td>Version 12.2.0.1.36</td>
<td>Latest patch p6880880_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>SnapCenter Service</td>
<td>Version</td>
<td>v2.3.1.2324</td>
</tr>
</tbody>
</table>

**Key factors for deployment consideration**

- **EC2 compute instances.** In these tests and validations, we used an AWS EC2 t2.xlarge instance type for the Oracle database compute instance. NetApp recommends using an M5 type EC2 instance as the compute instance for Oracle in production deployment because it is optimized for database workloads. You need to size the EC2 instance appropriately for the number of vCPUs and the amount of RAM based on actual workload requirements.

- **FSx storage HA clusters single- or multi-zone deployment.** In these tests and validations, we deployed an FSx HA cluster in a single AWS availability zone. For production deployment, NetApp recommends deploying an FSx HA pair in two different availability zones. An FSx HA cluster is always provisioned in a HA pair that is sync mirrored in a pair of active-passive file systems to provide storage-level redundancy. Multi-zone deployment further enhances high availability in the event of failure in a single AWS zone.

- **FSx storage cluster sizing.** An Amazon FSx for ONTAP storage file system provides up to 160,000 raw SSD IOPS, up to 4GBps throughput, and a maximum of 192TiB capacity. However, you can size the cluster in terms of provisioned IOPS, throughput, and the storage limit (minimum 1,024 GiB) based on your actual requirements at the time of deployment. The capacity can be adjusted dynamically on the fly without affecting application availability.

- **Oracle data and logs layout.** In our tests and validations, we deployed two ASM disk groups for data and logs respectively. Within the +DATA asm disk group, we provisioned four disks in a data NFS file system mount point. Within the +LOGS asm disk group, we provisioned two disks in a logs NFS file system mount point. For large database deployment, ASM disk groups can be built to span multiple FSx file systems with ASM NFS disks distributed through multiple NFS mount points anchored on FSx file systems. This particular setup is designed to meet database throughput over 4GBps throughput and 160,000 raw SSD IOPS requirement.

- **dNFS configuration.** dNFS is built into Oracle kernel and is known to dramatically increase Oracle database performance when Oracle is deployed to NFS storage. dNFS is packaged into Oracle binary but is not turned on by default. It should be turned on for any Oracle database deployment on NFS. For multiple FSx file systems deployment for large database, dNFS multi-path should be properly configured.

- **Oracle ASM redundancy level to use for each Oracle ASM disk group that you create.** Because FSx already mirrors the storage on the FSx cluster level, you should ONLY use External Redundancy, which means that the option does not allow Oracle ASM to mirror the contents of the disk group. This is
particularly important as NFS for Oracle database data storage requires HARD NFS mount option which is NOT desirable for mirroring ASM contents on the Oracle level.

- **Database backup.** NetApp provides a SaaS version of SnapCenter software service for database backup, restore, and clone in the cloud that is available through the NetApp BlueXP console UI. NetApp recommends implementing such a service to achieve fast (under a minute) SnapShot backup, quick (few minutes) database restore, and database cloning.

**Solution deployment**

The following section provides step-by-step deployment procedures.

**Prerequisites for deployment**

Deployment requires the following prerequisites.

1. An AWS account has been set up, and the necessary VPC and network segments have been created within your AWS account.

2. From the AWS EC2 console, you must deploy two EC2 Linux instances, one as the primary Oracle DB server and an optional alternative clone target DB server. See the architecture diagram in the previous section for more details about the environment setup. Also review the User Guide for Linux instances for more information.

3. From the AWS EC2 console, deploy Amazon FSx for ONTAP storage HA clusters to host the Oracle database volumes. If you are not familiar with the deployment of FSx storage, see the documentation Creating FSx for ONTAP file systems for step-by-step instructions.

4. Steps 2 and 3 can be performed using the following Terraform automation toolkit, which creates an EC2 instance named ora_01 and an FSx file system named fsx_01. Review the instruction carefully and change the variables to suit your environment before execution.

   ```bash
   git clone https://github.com/NetApp-Automation/na_aws_fsx_ec2_deploy.git
   ```

   Ensure that you have allocated at least 50G in EC2 instance root volume in order to have sufficient space to stage Oracle installation files.

**EC2 instance kernel configuration**
With the prerequisites provisioned, log into the EC2 instance as ec2-user and sudo to root user to configure the Linux kernel for Oracle installation.

1. Create a staging directory `/tmp/archive` folder and set the 777 permission.

   ```
   mkdir /tmp/archive
   chmod 777 /tmp/archive
   ```

2. Download and stage the Oracle binary installation files and other required rpm files to the `/tmp/archive` directory.

   See the following list of installation files to be stated in `/tmp/archive` on the EC2 instance.

   ```
   [ec2-user@ip-172-30-15-58 ~]$ ls -l /tmp/archive
   total 10537316
   -rw-rw-r--. 1 ec2-user ec2-user      19112 Mar 21 15:57 compat-libcap1-1.10-7.el7.x86_64.rpm
   -rw-rw-r--  1 ec2-user ec2-user 3059705302 Mar 21 22:01 LINUX.X64_193000_db_home.zip
   -rw-rw-r--  1 ec2-user ec2-user 2889184573 Mar 21 21:09 LINUX.X64_193000_grid_home.zip
   -rw-rw-r--. 1 ec2-user ec2-user     589145 Mar 21 15:56 netapp_linux_unified_host_utilities-7-1.x86_64.rpm
   -rw-rw-r--. 1 ec2-user ec2-user 31828 Mar 21 15:55 oracle-database-preinstall-19c-1.0-2.el8.x86_64.rpm
   -rw-rw-r--. 1 ec2-user ec2-user 2872741741 Mar 21 22:31 p34762026_190000_Linux-x86-64.zip
   -rw-rw-r--  1 ec2-user ec2-user 1843577895 Mar 21 22:32 p34765931_190000_Linux-x86-64.zip
   -rw-rw-r--  1 ec2-user ec2-user 124347218 Mar 21 22:33 p6880880_190000_Linux-x86-64.zip
   -rw-rw-r--  1 ec2-user ec2-user 257136 Mar 22 16:25 policycoreutils-python-utils-2.9-9.el8.noarch.rpm
   ```

3. Install Oracle 19c preinstall RPM, which satisfies most kernel configuration requirements.

   ```
   yum install /tmp/archive/oracle-database-preinstall-19c-1.0-2.el8.x86_64.rpm
   ```

4. Download and install the missing `compat-libcap1` in Linux 8.
yum install /tmp/archive/compat-libcap1-1.10-7.el7.x86_64.rpm

5. From NetApp, download and install NetApp host utilities.

yum install /tmp/archive/netapp_linux_unified_host_utilities-7-1.x86_64.rpm

6. Install policycoreutils-python-utils, which is not available in the EC2 instance.

yum install /tmp/archive/policycoreutils-python-utils-2.9-9.el8.noarch.rpm

7. Install open JDK version 1.8.

yum install java-1.8.0-openjdk.x86_64

8. Install nfs-utils.

yum install nfs-utils

9. Disable transparent hugepages in the current system.

    echo never > /sys/kernel/mm/transparent_hugepage/enabled
    echo never > /sys/kernel/mm/transparent_hugepage/defrag

Add the following lines in /etc/rc.local to disable transparent_hugepage after reboot:

    # Disable transparent hugepages
    if test -f /sys/kernel/mm/transparent_hugepage/enabled;
    then
        echo never > /sys/kernel/mm/transparent_hugepage/enabled
    fi
    if test -f /sys/kernel/mm/transparent_hugepage/defrag;
    then
        echo never > /sys/kernel/mm/transparent_hugepage/defrag
    fi

10. Disable selinux by changing SELINUX=enforcing to SELINUX=disabled. You must reboot the host to make the change effective.
11. Add the following lines to limit.conf to set the file descriptor limit and stack size without quotes " ".

```
vi /etc/security/limits.conf

"*
  hard  nofile  65536"
"*
  soft  stack  10240"
```

12. Add swap space to EC2 instance by following this instruction: How do I allocate memory to work as swap space in an Amazon EC2 instance by using a swap file? The exact amount of space to add depends on the size of RAM up to 16G.

13. Add the ASM group to be used for the asm sysasm group

```
groupadd asm
```

14. Modify the oracle user to add ASM as a secondary group (the oracle user should have been created after Oracle preinstall RPM installation).

```
usermod -a -G asm oracle
```

15. Reboot the EC2 instance.

Provision and export NFS volumes to be mounted to EC2 instance host
Provision three volumes from the command line by login to FSx cluster via ssh as fsxadmin user with FSx cluster management IP to host the Oracle database binary, data, and logs files.

1. Log into the FSx cluster through SSH as the fsxadmin user.

   ```bash
   ssh fsxadmin@172.30.15.53
   ```

2. Execute the following command to create a volume for the Oracle binary.

   ```bash
   vol create -volume ora_01_biny -aggregate aggr1 -size 50G -state online -type RW -junction-path /ora_01_biny -snapshot-policy none -tiering-policy snapshot-only
   ```

3. Execute the following command to create a volume for Oracle data.

   ```bash
   vol create -volume ora_01_data -aggregate aggr1 -size 100G -state online -type RW -junction-path /ora_01_data -snapshot-policy none -tiering-policy snapshot-only
   ```

4. Execute the following command to create a volume for Oracle logs.

   ```bash
   vol create -volume ora_01_logs -aggregate aggr1 -size 100G -state online -type RW -junction-path /ora_01_logs -snapshot-policy none -tiering-policy snapshot-only
   ```

5. Validate the DB volumes created.

   ```bash
   vol show
   ```

   This is expected to return:
### Database storage configuration

```plaintext
FsxId02ad7bf3476b741df:~> vol show

FsxId06c3c8b2a7bd56458:~> vol show

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Volume</th>
<th>Aggregate</th>
<th>State</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>svm_ora</td>
<td>ora_01_biny</td>
<td>aggr1</td>
<td>online</td>
<td>RW</td>
<td>50GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>47.50GB</td>
</tr>
<tr>
<td>svm_ora</td>
<td>ora_01_data</td>
<td>aggr1</td>
<td>online</td>
<td>RW</td>
<td>100GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>95.00GB</td>
</tr>
<tr>
<td>svm_ora</td>
<td>ora_01_logs</td>
<td>aggr1</td>
<td>online</td>
<td>RW</td>
<td>100GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>95.00GB</td>
</tr>
<tr>
<td>svm_ora</td>
<td>svm_ora_root</td>
<td>aggr1</td>
<td>online</td>
<td>RW</td>
<td>1GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>972.1MB</td>
</tr>
</tbody>
</table>

4 entries were displayed.
```
Now, import and set up the FSx storage for the Oracle grid infrastructure and database installation on the EC2 instance host.

1. Log into the EC2 instance via SSH as the ec2-user with your SSH key and EC2 instance IP address.

   ssh -i ora_01.pem ec2-user@172.30.15.58

2. Create /u01 directory to mount Oracle binary file system

   sudo mkdir /u01

3. Mount the binary volume to /u01, changed to your FSx NFS lif IP address. If you deployed FSx cluster via NetApp automation toolkit, FSx virtual storage server NFS lif IP address will be listed in the output at the end of resources provision execution. Otherwise, it can be retrieved from AWS FSx console UI.

   sudo mount -t nfs 172.30.15.19:/ora_01_biny /u01 -o rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536

4. Change /u01 mount point ownership to the Oracle user and it’s associated primary group.

   sudo chown oracle:oinstall /u01

5. Create /oradata directory to mount Oracle data file system

   sudo mkdir /oradata

6. Mount the data volume to /oradata, changed to your FSx NFS lif IP address

   sudo mount -t nfs 172.30.15.19:/ora_01_data /oradata -o rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536

7. Change /oradata mount point ownership to the Oracle user and it’s associated primary group.

   sudo chown oracle:oinstall /oradata

8. Create /oralogs directory to mount Oracle logs file system

   sudo mkdir /oralogs
9. Mount the log volume to `/oralogs`, changed to your FSx NFS lif IP address

```
sudo mount -t nfs 172.30.15.19:/ora_01_logs /oralogs -o rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536
```

10. Change `/oralogs` mount point ownership to the Oracle user and it's associated primary group.

```
sudo chown oracle:oinstall /oralogs
```

11. Add a mount point to `/etc/fstab`.

```
sudo vi /etc/fstab
```

Add the following line.

```
172.30.15.19:/ora_01_biny       /u01            nfs
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536   0
0
172.30.15.19:/ora_01_data       /oradata        nfs
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536   0
0
172.30.15.19:/ora_01_logs       /oralogs        nfs
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536   0
0
```

12. `sudo` to oracle user, create asm folders to store asm disk files

```
sudo su
su - oracle
mkdir /oradata/asm
mkdir /oralogs/asm
```

13. As the oracle user, create asm data disk files, change the count to match to the disk size with block size.
dd if=/dev/zero of=/oradata/asm/nfs_data_disk01 bs=1M count=20480 oflag=direct
dd if=/dev/zero of=/oradata/asm/nfs_data_disk02 bs=1M count=20480 oflag=direct
dd if=/dev/zero of=/oradata/asm/nfs_data_disk03 bs=1M count=20480 oflag=direct
dd if=/dev/zero of=/oradata/asm/nfs_data_disk04 bs=1M count=20480 oflag=direct

14. As the root user, change data disk file permission to 640

    chmod 640 /oradata/asm/*

15. As the oracle user, create asm logs disk files, change to count to match to the disk size with block size.

    dd if=/dev/zero of=/oralogs/asm/nfs_logs_disk01 bs=1M count=40960 oflag=direct
    dd if=/dev/zero of=/oralogs/asm/nfs_logs_disk02 bs=1M count=40960 oflag=direct

16. As the root user, change logs disk file permission to 640

    chmod 640 /oralogs/asm/*

17. Reboot the EC2 instance host.

Oracle grid infrastructure installation
1. Log into the EC2 instance as the ec2-user via SSH and enable password authentication by uncommenting `PasswordAuthentication yes` and then commenting out `PasswordAuthentication no`.

```
sudo vi /etc/ssh/sshd_config
```

2. Restart the sshd service.

```
sudo systemctl restart sshd
```

3. Reset the Oracle user password.

```
sudo passwd oracle
```

4. Log in as the Oracle Restart software owner user (oracle). Create an Oracle directory as follows:

```
mkdir -p /u01/app/oracle
mkdir -p /u01/app/oraInventory
```

5. Change the directory permission setting.

```
chmod -R 775 /u01/app
```

6. Create a grid home directory and change to it.

```
mkdir -p /u01/app/oracle/product/19.0.0/grid
cd /u01/app/oracle/product/19.0.0/grid
```

7. Unzip the grid installation files.

```
unzip -q /tmp/archive/LINUX.X64_193000_grid_home.zip
```

8. From grid home, delete the OPatch directory.

```
rm -rf OPatch
```

9. From grid home, copy `p6880880_190000_Linux-x86-64.zip` to the grid_home, and then unzip it.
cp /tmp/archive/p6880880_190000_Linux-x86-64.zip .
unzip p6880880_190000_Linux-x86-64.zip

10. From grid home, revise cv/admin/cvu_config, uncomment and replace
    CV_ASSUME_DISTID=OEL5 with CV_ASSUME_DISTID=OL7.

    vi cv/admin/cvu_config

11. Prepare a gridsetup.rsp file for silent installation and place the rsp file in the /tmp/archive directory. The rsp file should cover sections A, B, and G with the following information:

    INVENTORY_LOCATION=/u01/app/oraInventory
    oracle.install.option=HA_CONFIG
    ORACLE_BASE=/u01/app/oracle
    oracle.install.asm.OSDBA=dba
    oracle.install.asm.OSOPER=oper
    oracle.install.asm.OSASM=asm
    oracle.install.asm.SYSASMPassword="SetPWD"
    oracle.install.asm.diskGroup.name=DATA
    oracle.install.asm.diskGroup.redundancy=EXTERNAL
    oracle.install.asm.diskGroup.AUSize=4
    oracle.install.asm.diskGroup.disks=/oradata/asm/*,/oralogs/asm/*
    oracle.install.asm.diskGroup.diskDiscoveryString=/oradata/asm/nfs_data_disk01,/oradata/asm/nfs_data_disk02,/oradata/asm/nfs_data_disk03,/oradata/asm/nfs_data_disk04
    oracle.install.asm.monitorPassword="SetPWD"
    oracle.install.asm.configureAFD=false

12. Log into the EC2 instance as the root user.
13. Install cvuqdisk-1.0.10-1.rpm.

    rpm -ivh /u01/app/oracle/product/19.0.0/grid/cv/rpm/cvuqdisk-1.0.10-1.rpm

14. Log into the EC2 instance as the Oracle user and extract the patch in the /tmp/archive folder.

    unzip p34762026_190000_Linux-x86-64.zip

15. From grid home /u01/app/oracle/product/19.0.0/grid and as the oracle user, launch gridSetup.sh for grid infrastructure installation.
Ignore the warnings about wrong groups for grid infrastructure. We are using a single Oracle user to manage Oracle Restart, so this is expected.

16. As root user, execute the following script(s):

```
/u01/app/oraInventory/orainstRoot.sh
/u01/app/oracle/product/19.0.0/grid/root.sh
```

17. As the Oracle user, execute the following command to complete the configuration:

```
/u01/app/oracle/product/19.0.0/grid/gridSetup.sh -executeConfigTools -responseFile /tmp/archive/gridsetup.rsp -silent
```

18. As the Oracle user, create the LOGS disk group.

```
bin/asmca -silent -sysAsmPassword 'yourPWD' -asmsnmpPassword 'yourPWD' -createDiskGroup -diskGroupName LOGS -disk '/oralogs/asm/nfs_logs_disk*' -redundancy EXTERNAL -au_size 4
```

19. As the Oracle user, validate grid services after installation configuration.
```
bin/crsctl stat res -t
+
<table>
<thead>
<tr>
<th>Name</th>
<th>Target</th>
<th>State</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>State details</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local Resources</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ora.DATA.dg</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
<tr>
<td>oradata</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
<tr>
<td>ora.LOGIN.dg</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
<tr>
<td>ora.LISTENER.lsnr</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
<tr>
<td>oralisten</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
<tr>
<td>ora.LOGS.dg</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
<tr>
<td>oralog</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
<tr>
<td>ora.asm</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
<tr>
<td>orasm</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
<tr>
<td>Started,STABLE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster Resources</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ora.cssd</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
<tr>
<td>oracssd</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
<tr>
<td>ora.diskmon</td>
<td>OFFLINE</td>
<td>OFFLINE</td>
<td></td>
</tr>
<tr>
<td>oradiskmon</td>
<td>OFFLINE</td>
<td>OFFLINE</td>
<td></td>
</tr>
<tr>
<td>ora.driver.afd</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
<tr>
<td>oradriver.afd</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
<tr>
<td>ora.evmd</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
</tbody>
</table>
```

Oracle database installation

1001
1. Log in as the Oracle user and unset $ORACLE_HOME and $ORACLE_SID if it is set.

   ```bash
   unset ORACLE_HOME
   unset ORACLE_SID
   ```

2. Create the Oracle DB home directory and change to it.

   ```bash
   mkdir /u01/app/oracle/product/19.0.0/db1
   cd /u01/app/oracle/product/19.0.0/db1
   ```

3. Unzip the Oracle DB installation files.

   ```bash
   unzip -q /tmp/archive/LINUX.X64_193000_db_home.zip
   ```

4. From the DB home, delete the OPatch directory.

   ```bash
   rm -rf OPatch
   ```

5. From DB home, copy `p6880880_190000_Linux-x86-64.zip` to grid_home, and then unzip it.

   ```bash
   cp /tmp/archive/p6880880_190000_Linux-x86-64.zip .
   unzip p6880880_190000_Linux-x86-64.zip
   ```

6. From DB home, revise `cv/admin/cvu_config`, and uncomment and replace `CV_ASSUME_DISTID=OEL5` with `CV_ASSUME_DISTID=OL7`.

   ```bash
   vi cv/admin/cvu_config
   ```

7. From the `/tmp/archive` directory, unpack the DB 19.18 RU patch.

   ```bash
   unzip p34765931_190000_Linux-x86-64.zip
   ```

8. Prepare the DB silent install rsp file in `/tmp/archive/dbinstall.rsp` directory with the following values:
9. From db1 home /u01/app/oracle/product/19.0.0/db1, execute silent software-only DB installation.

   ./runInstaller -applyRU /tmp/archive/34765931/ -silent
   -ignorePrereqFailure -responseFile /tmp/archive/dbinstall.rsp

10. As root user, run the root.sh script after software-only installation.

    /u01/app/oracle/product/19.0.0/db1/root.sh

11. As Oracle user, create the dbca.rsp file with the following entries:
gdbName=db1.demo.netapp.com
sid=db1
createAsContainerDatabase=true
numberOfPDBs=3
pdbName=db1_pdb
useLocalUndoForPDBs=true
pdbAdminPassword="yourPWD"
templateName=General_Purpose.dbc
sysPassword="yourPWD"
systemPassword="yourPWD"
dbsnmpPassword="yourPWD"
stORAGEType=ASM
diskGroupName=DATA
characterSet=AL32UTF8
nationalCharacterSet=AL16UTF16
listeners=LISTENER
databaseType=MULTIPURPOSE
automaticMemoryManagement=false
totalMemory=8192

Set the total memory based on available memory in EC2 instance host. Oracle allocates 75% of totalMemory to DB instance SGA or buffer cache.

12. As Oracle user, launch DB creation with dbca.
bin/dbca -silent -createDatabase -responseFile /tmp/archive/dbca.rsp

output:
Prepare for db operation
7% complete
Registering database with Oracle Restart
11% complete
Copying database files
33% complete
Creating and starting Oracle instance
35% complete
38% complete
42% complete
45% complete
48% complete
Completing Database Creation
53% complete
55% complete
56% complete
Creating Pluggable Databases
60% complete
64% complete
69% complete
78% complete
Executing Post Configuration Actions
100% complete
Database creation complete. For details check the logfiles at:
  /u01/app/oracle/cfgtoollogs/dbca/db1.
Database Information:
Global Database Name:db1.demo.netapp.com
System Identifier(SID):db1
Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/db1/db1.log"
for further details.

13. As Oracle user, validate Oracle Restart HA services after DB creation.
```bash
[oracle@ip-172-30-15-58 db1]$ ../grid/bin/crsctl stat res -t
--------------------------------------------------------------------
------------
<table>
<thead>
<tr>
<th>Name</th>
<th>Target</th>
<th>State</th>
<th>Server</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>details</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local Resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ora.DATA.dg</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.LISTENER.lsnr</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.LOGS.dg</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.asm</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>Started,STABLE</td>
</tr>
<tr>
<td>ora.ons</td>
<td>OFFLINE</td>
<td>OFFLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster Resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ora.cssd</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.db1.db</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>/db1,STABLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ora.diskmon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>OFFLINE</td>
<td>OFFLINE</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.evmd</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
</tbody>
</table>
--------------------------------------------------------------------

[oracle@ip-172-30-15-58 db1]$  

14. Set the Oracle user .bash_profile.
```
vi ~/.bash_profile

15. Add following entries:

```
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db1
export ORACLE_SID=db1
export PATH=$PATH:$ORACLE_HOME/bin
alias asm='export
ORACLE_HOME=/u01/app/oracle/product/19.0.0/grid;export
ORACLE_SID=+ASM;export PATH=$PATH:$ORACLE_HOME/bin'
```

16. Validate the CDB/PDB created.

```
. ~/.bash_profile
sqlplus / as sysdba
SQL> select name, open_mode from v$database;
NAME      OPEN_MODE
DB1       READ WRITE
SQL> select name from v$datafile;
NAME
+DATA/DB1/DATAFILE/system.256.1132176177
+DATA/DB1/DATAFILE/sysaux.257.1132176221
+DATA/DB1/DATAFILE/undotbs1.258.1132176247
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/system.265.1132177009
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/sysaux.266.1132177009
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/undotbs1.267.1132177009
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/system.271.1132177853
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/sysaux.272.1132177853
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/undotbs1.270.1132177853
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/users.274.1132177853
```
17. As oracle user, change to Oracle database home directory /u01/app/oracle/product/19.0.0/db1 and Enable dNFS

    cd /u01/app/oracle/product/19.0.0/db1
    mkdir rdbms/lib/odm
    cp lib/libnfsodm19.so rdbms/lib/odm/

18. Configure oranfstab file in ORACLE_HOME
vi $ORACLE_HOME/dbs/oranfstab

add following entries:

server: fsx_01
local: 172.30.15.58 path: 172.30.15.19
nfs_version: nfsv3
export: /ora_01_biny mount: /u01
export: /ora_01_data mount: /oradata
export: /ora_01_logs mount: /oralogs

19. As oracle user, login to database from sqlplus and set the DB recovery size and location to the +LOGS disk group.

. ~/.bash_profile

sqlplus / as sysdba

alter system set db_recovery_file_dest_size = 80G scope=both;
alter system set db_recovery_file_dest = '+LOGS' scope=both;

20. Enable archive log mode and reboot Oracle DB instance

shutdown immediate;
startup mount;
alter database archivelog;
alter database open;
alter system switch logfile;

21. Validate DB log mode and dNFS after instance reboot
SELECT name, log_mode FROM v$database;

NAME   LOG_MODE
------- ---------
DB1     ARCHIVELOG

SELECT svrname, dirname FROM v$dnfs_servers;

SVRNAME             
-------------------
DIRNAME             
-------------------
fsx_01              
/ora_01_data        
fsx_01              
/ora_01_biny        
fsx_01              
/ora_01_logs

22. Validate Oracle ASM

[oracle@ip-172-30-15-58 db1]$ asm
[oracle@ip-172-30-15-58 db1]$ sqlplus / as sysasm

SQL*Plus: Release 19.0.0.0.0 - Production on Tue May 9 20:39:39 2023
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0

SQL> set lin 200
SQL> col path form a30
SQL> select name, path, header_status, mount_status, state from v$asm_disk;

NAME   PATH
------- ---
1010

This completes Oracle 19c version 19.18 Restart deployment on an Amazon FSx for ONTAP and EC2 compute instance with NFS/ASM. If desired, NetApp recommends relocating the Oracle control file and online log files to the +LOGS disk group.

Automated deployment option

NetApp will release a fully automated solution deployment toolkit with Ansible to facilitate the implementation of this solution. Please check back for the availability of the toolkit. After it is released, a link will be posted here.

Oracle Database backup, restore, and clone with SnapCenter Service

At this moment, Oracle database with NFS and ASM storage option is only supported by traditional SnapCenter Server UI tool See Hybrid Cloud Database Solutions with SnapCenter for details on Oracle database backup, restore, and clone with NetApp SnapCenter UI tool.
Where to find additional information

To learn more about the information described in this document, review the following documents and/or websites:

• Installing Oracle Grid Infrastructure for a Standalone Server with a New Database Installation

• Installing and Configuring Oracle Database Using Response Files

• Amazon FSx for NetApp ONTAP
  https://aws.amazon.com/fsx/netapp-ontap/

• Amazon EC2
  https://aws.amazon.com/pm/ec2/?trk=36c6da98-7b20-48fa-8225-4784bcd9843&sc_channel=ps&s_kwcid=AL!4422!3!467723097970le!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARIsAJzSrOqwQrghn6I71jiWzSeaT9UUh1-vY-VfhJixF-sxnv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:G:s&s_kwcid=AL!4422!3!467723097970le!!g!!aws%20ec2

TR-4965: Oracle Database Deployment and Protection in AWS FSx/EC2 with iSCSI/ASM

Allen Cao, Niyaz Mohamed, NetApp

Purpose

ASM (Automatic Storage Management) is a popular Oracle storage volume manager employed in many Oracle installations. It is also Oracle’s recommended storage management solution. It provides an alternative to conventional volume managers and file systems. Since Oracle version 11g, ASM packaged with grid infrastructure rather than a database. As a result, in order to utilize Oracle ASM for storage management without RAC, you must install Oracle grid infrastructure in a standalone server, also known as Oracle Restart. Doing so certainly adds more complexity in Oracle database deployment. However, as the name implies, when Oracle deployed in Restart mode, failed Oracle services restarted automatically by grid infrastructure or after a host reboot without user intervention, which provides a certain degree of high availability or HA functionality.

In this documentation, we demonstrate how to deploy an Oracle database with the iSCSI protocol and Oracle ASM in an Amazon FSx for ONTAP storage environment with EC2 compute instances. We also demonstrate how to use the NetApp SnapCenter service through the NetApp BlueXP console to backup, restore, and clone your Oracle database for dev/test or other use cases for storage-efficient database operation in the AWS public cloud.

This solution addresses the following use cases:

• Oracle database deployment in Amazon FSx for ONTAP storage and EC2 compute instances with iSCSI/ASM
• Testing and validating an Oracle workload in the public AWS cloud with iSCSI/ASM
• Testing and validating Oracle database Restart functionalities deployed in AWS

**Audience**

This solution is intended for the following people:

• A DBA who would like to deploy Oracle in an AWS public cloud with iSCSI/ASM.
• A database solution architect who would like to test Oracle workloads in the AWS public cloud.
• The storage administrator who would like to deploy and manage an Oracle database deployed to AWS FSx storage.
• The application owner who would like to stand up an Oracle database in AWS FSx/EC2.

**Solution test and validation environment**

The testing and validation of this solution was performed in an AWS FSx and EC2 environment that might not match the final deployment environment. For more information, see the section [Key Factors for Deployment Consideration].

**Architecture**

**Oracle Deployment in AWS FSx/EC2 with iSCSI/ASM**

**Hardware and software components**

<table>
<thead>
<tr>
<th>Hardware</th>
<th>FSx ONTAP storage</th>
<th>EC2 instance for compute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current version offered by AWS</td>
<td>One FSx HA cluster in the same VPC and availability zone</td>
<td></td>
</tr>
<tr>
<td>t2.xlarge/4vCPU/16G</td>
<td>Two EC2 T2 xlarge EC2 instances, one as primary DB server and the other as a clone DB server</td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td>Version</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>RedHat Linux</td>
<td>RHEL-8.6.0_HVM-20220503-x86_64-2-Hourly2-GP2</td>
<td>Deployed RedHat subscription for testing</td>
</tr>
<tr>
<td>Oracle Grid Infrastructure</td>
<td>Version 19.18</td>
<td>Applied RU patch p34762026_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle Database</td>
<td>Version 19.18</td>
<td>Applied RU patch p34765931_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle OPatch</td>
<td>Version 12.2.0.1.36</td>
<td>Latest patch p6880880_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>SnapCenter Service</td>
<td>Version</td>
<td>v2.3.1.2324</td>
</tr>
</tbody>
</table>

**Key factors for deployment consideration**

- **EC2 compute instances.** In these tests and validations, we used an AWS EC2 t2.xlarge instance type for the Oracle database compute instance. NetApp recommends using an M5 type EC2 instance as the compute instance for Oracle in production deployment because it is optimized for database workloads. You need to size the EC2 instance appropriately for the number of vCPUs and the amount of RAM based on actual workload requirements.

- **FSx storage HA clusters single- or multi-zone deployment.** In these tests and validations, we deployed an FSx HA cluster in a single AWS availability zone. For production deployment, NetApp recommends deploying an FSx HA pair in two different availability zones. An FSx HA cluster is always provisioned in a HA pair that is sync mirrored in a pair of active-passive file systems to provide storage-level redundancy. Multi-zone deployment further enhances high availability in the event of failure in a single AWS zone.

- **FSx storage cluster sizing.** An Amazon FSx for ONTAP storage file system provides up to 160,000 raw SSD IOPS, up to 4GBps throughput, and a maximum of 192TiB capacity. However, you can size the cluster in terms of provisioned IOPS, throughput, and the storage limit (minimum 1,024 GiB) based on your actual requirements at the time of deployment. The capacity can be adjusted dynamically on the fly without affecting application availability.

- **Oracle data and logs layout.** In our tests and validations, we deployed two ASM disk groups for data and logs respectively. Within the +DATA asm disk group, we provisioned four LUNs in a data volume. Within the +LOGS asm disk group, we provisioned two LUNs in a logs volume. In general, multiple LUNs laid out within an Amazon FSx for ONTAP volume provides better performance.

- **iSCSI configuration.** The EC2 instance database server connects to FSx storage with the iSCSI protocol. EC2 instances generally deploy with a single network interface or ENI. The single NIC interface carries both iSCSI and application traffic. It is important to gauge the Oracle database peak I/O throughput requirement by carefully analyzing the Oracle AWR report in order to choose a right EC2 compute instance that meets both application and iSCSI traffic-throughput requirements. NetApp also recommends allocating four iSCSI connections to both FSx iSCSI endpoints with multipath properly configured.

- **Oracle ASM redundancy level to use for each Oracle ASM disk group that you create.** Because FSx already mirrors the storage on the FSx cluster level, you should use External Redundancy, which means that the option does not allow Oracle ASM to mirror the contents of the disk group.

- **Database backup.** NetApp provides a SaaS version of SnapCenter software service for database backup, restore, and clone in the cloud that is available through the NetApp BlueXP console UI. NetApp recommends implementing such a service to achieve fast (under a minute) SnapShot backup, quick (few minutes) database restore, and database cloning.
Solution deployment

The following section provides step-by-step deployment procedures.

Prerequisites for deployment

Deployment requires the following prerequisites.

1. An AWS account has been set up, and the necessary VPC and network segments have been created within your AWS account.

2. From the AWS EC2 console, you must deploy two EC2 Linux instances, one as the primary Oracle DB server and an optional alternative clone target DB server. See the architecture diagram in the previous section for more details about the environment setup. Also review the User Guide for Linux instances for more information.

3. From the AWS EC2 console, deploy Amazon FSx for ONTAP storage HA clusters to host the Oracle database volumes. If you are not familiar with the deployment of FSx storage, see the documentation Creating FSx for ONTAP file systems for step-by-step instructions.

4. Steps 2 and 3 can be performed using the following Terraform automation toolkit, which creates an EC2 instance named ora_01 and an FSx file system named fsx_01. Review the instruction carefully and change the variables to suit your environment before execution.

```bash
git clone https://github.com/NetApp-Automation/na_aws_fsx_ec2_deploy.git
```

Ensure that you have allocated at least 50G in EC2 instance root volume in order to have sufficient space to stage Oracle installation files.

EC2 instance kernel configuration
With the prerequisites provisioned, log into the EC2 instance as ec2-user and sudo to root user to configure the Linux kernel for Oracle installation.

1. Create a staging directory `/tmp/archive` folder and set the `777` permission.

   ```
   mkdir /tmp/archive
   chmod 777 /tmp/archive
   ```

2. Download and stage the Oracle binary installation files and other required rpm files to the `/tmp/archive` directory.

   See the following list of installation files to be stated in `/tmp/archive` on the EC2 instance.

   ```
   [ec2-user@ip-172-30-15-58 ~]$ ls -l /tmp/archive
   total 10537316
   -rw-rw-r--. 1 ec2-user ec2-user      19112 Mar 21 15:57 compat-libcap1-1.10-7.el7.x86_64.rpm
   -rw-rw-r--  1 ec2-user ec2-user 3059705302 Mar 21 22:01 LINUX.X64_193000_db_home.zip
   -rw-rw-r--  1 ec2-user ec2-user 2889184573 Mar 21 21:09 LINUX.X64_193000_grid_home.zip
   -rw-rw-r--. 1 ec2-user ec2-user     589145 Mar 21 15:56 netapp_linux_unified_host_utilities-7-1.x86_64.rpm
   -rw-rw-r--. 1 ec2-user ec2-user 31828 Mar 21 15:55 oracle-database-preinstall-19c-1.0-2.el8.x86_64.rpm
   -rw-rw-r--. 1 ec2-user ec2-user 2872741741 Mar 21 22:31 p34762026_190000_Linux-x86-64.zip
   -rw-rw-r--. 1 ec2-user ec2-user 1843577895 Mar 21 22:32 p34765931_190000_Linux-x86-64.zip
   -rw-rw-r--. 1 ec2-user ec2-user 124347218 Mar 21 22:33 p6880880_190000_Linux-x86-64.zip
   -rw-r-r--  1 ec2-user ec2-user     257136 Mar 22 16:25 policycoreutils-python-utils-2.9-9.el8.noarch.rpm
   ```

3. Install Oracle 19c preinstall RPM, which satisfies most kernel configuration requirements.

   ```
   yum install /tmp/archive/oracle-database-preinstall-19c-1.0-2.el8.x86_64.rpm
   ```

4. Download and install the missing `compat-libcap1` in Linux 8.
yum install /tmp/archive/compat-libcap1-1.10-7.el7.x86_64.rpm

5. From NetApp, download and install NetApp host utilities.

yum install /tmp/archive/netapp_linux_unified_host_utilities-7-1.x86_64.rpm

6. Install policycoreutils-python-utils, which is not available in the EC2 instance.

yum install /tmp/archive/policycoreutils-python-utils-2.9-9.el8.noarch.rpm

7. Install open JDK version 1.8.

yum install java-1.8.0-openjdk.x86_64

8. Install iSCSI initiator utils.

yum install iscsi-initiator-utils

9. Install sg3_utils.

yum install sg3_utils

10. Install device-mapper-multipath.

yum install device-mapper-multipath

11. Disable transparent hugepages in the current system.

    echo never > /sys/kernel/mm/transparent_hugepage/enabled
    echo never > /sys/kernel/mm/transparent_hugepage/defrag

Add the following lines in /etc/rc.local to disable transparent_hugepage after reboot:
12. Disable selinux by changing `SELINUX=enforcing` to `SELINUX=disabled`. You must reboot the host to make the change effective.

```
vi /etc/sysconfig/selinux
```

13. Add the following lines to `limit.conf` to set the file descriptor limit and stack size without quotes " ".

```
vi /etc/security/limits.conf

"*" hard nofile 65536
"*" soft stack 10240
```

14. Add swap space to EC2 instance by following this instruction: How do I allocate memory to work as swap space in an Amazon EC2 instance by using a swap file? The exact amount of space to add depends on the size of RAM up to 16G.

15. Change `node.session.timeo.replacement_timeout` in the `iscsi.conf` configuration file from 120 to 5 seconds.

```
vi /etc/iscsi/iscsid.conf
```

16. Enable and start the iSCSI service on the EC2 instance.

```
systemctl enable iscsid
systemctl start iscsid
```

17. Retrieve the iSCSI initiator address to be used for database LUN mapping.

```
cat /etc/iscsi/initiatorname.iscsi
```

18. Add the ASM group to be used for the asm sysasm group.
19. Modify the oracle user to add ASM as a secondary group (the oracle user should have been created after Oracle preinstall RPM installation).

```bash
usermod -a -G asm oracle
```

20. Stop and disable Linux firewall if it is active.

```bash
systemctl stop firewalld
systemctl disable firewalld
```

21. Reboot the EC2 instance.

Provision and map database volumes and LUNs to the EC2 instance host
Provision three volumes from the command line by login to FSx cluster via ssh as fsxadmin user with FSx cluster management IP to host the Oracle database binary, data, and logs files.

1. Log into the FSx cluster through SSH as the fsxadmin user.

   ```
   ssh fsxadmin@172.30.15.53
   ```

2. Execute the following command to create a volume for the Oracle binary.

   ```
   vol create -volume ora_01_biny -aggregate aggr1 -size 50G -state
   online -type RW -snapshot-policy none -tiering-policy snapshot-only
   ```

3. Execute the following command to create a volume for Oracle data.

   ```
   vol create -volume ora_01_data -aggregate aggr1 -size 100G -state
   online -type RW -snapshot-policy none -tiering-policy snapshot-only
   ```

4. Execute the following command to create a volume for Oracle logs.

   ```
   vol create -volume ora_01_logs -aggregate aggr1 -size 100G -state
   online -type RW -snapshot-policy none -tiering-policy snapshot-only
   ```

5. Create a binary LUN within the database binary volume.

   ```
   lun create -path /vol/ora_01_biny/ora_01_biny_01 -size 40G -ostype
   linux
   ```

6. Create data LUNs within the database data volume.

   ```
   lun create -path /vol/ora_01_data/ora_01_data_01 -size 20G -ostype
   linux
   
   lun create -path /vol/ora_01_data/ora_01_data_02 -size 20G -ostype
   linux
   
   lun create -path /vol/ora_01_data/ora_01_data_03 -size 20G -ostype
   linux
   
   lun create -path /vol/ora_01_data/ora_01_data_04 -size 20G -ostype
   linux
   ```
7. Create log LUNs within the database logs volume.

```
lun create -path /vol/ora_01_logs/ora_01_logs_01 -size 40G -ostype linux
lun create -path /vol/ora_01_logs/ora_01_logs_02 -size 40G -ostype linux
```

8. Create an igroup for the EC2 instance with the initiator retrieved from step 14 of the EC2 kernel configuration above.

```
igroup create -igroup ora_01 -protocol iscsi -ostype linux
```

9. Map the LUNs to the igroup created above. Increment the LUN ID sequentially for each additional LUN within a volume.

```
lun map -path /vol/ora_01_biny/ora_01_biny_01 -igroup ora_01 -vserver svm_ora -lun-id 0
lun map -path /vol/ora_01_data/ora_01_data_01 -igroup ora_01 -vserver svm_ora -lun-id 1
lun map -path /vol/ora_01_data/ora_01_data_02 -igroup ora_01 -vserver svm_ora -lun-id 2
lun map -path /vol/ora_01_data/ora_01_data_03 -igroup ora_01 -vserver svm_ora -lun-id 3
lun map -path /vol/ora_01_data/ora_01_data_04 -igroup ora_01 -vserver svm_ora -lun-id 4
lun map -path /vol/ora_01_logs/ora_01_logs_01 -igroup ora_01 -vserver svm_ora -lun-id 5
lun map -path /vol/ora_01_logs/ora_01_logs_02 -igroup ora_01 -vserver svm_ora -lun-id 6
```

10. Validate the LUN mapping.

```
mapping show
```

This is expected to return:
<table>
<thead>
<tr>
<th>Vserver</th>
<th>Path</th>
<th>Igroup</th>
<th>LUN ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>svm_ora</td>
<td>/vol/ora_01_biny/ora_01_biny_01</td>
<td>ora_01</td>
<td>0</td>
</tr>
<tr>
<td>iscsi</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>svm_ora</td>
<td>/vol/ora_01_data/ora_01_data_01</td>
<td>ora_01</td>
<td>1</td>
</tr>
<tr>
<td>iscsi</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>svm_ora</td>
<td>/vol/ora_01_data/ora_01_data_02</td>
<td>ora_01</td>
<td>2</td>
</tr>
<tr>
<td>iscsi</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>svm_ora</td>
<td>/vol/ora_01_data/ora_01_data_03</td>
<td>ora_01</td>
<td>3</td>
</tr>
<tr>
<td>iscsi</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>svm_ora</td>
<td>/vol/ora_01_data/ora_01_data_04</td>
<td>ora_01</td>
<td>4</td>
</tr>
<tr>
<td>iscsi</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>svm_ora</td>
<td>/vol/ora_01_logs/ora_01_logs_01</td>
<td>ora_01</td>
<td>5</td>
</tr>
<tr>
<td>iscsi</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>svm_ora</td>
<td>/vol/ora_01_logs/ora_01_logs_02</td>
<td>ora_01</td>
<td>6</td>
</tr>
</tbody>
</table>

Database storage configuration
Now, import and set up the FSx storage for the Oracle grid infrastructure and database installation on the EC2 instance host.

1. Log into the EC2 instance via SSH as the ec2-user with your SSH key and EC2 instance IP address.

   ssh -i ora_01.pem ec2-user@172.30.15.58

2. Discover the FSx iSCSI endpoints using either SVM iSCSI IP address. Then change to your environment-specific portal address.

   sudo iscsiadm --mode discovery --op update --type sendtargets --portal 172.30.15.51

3. Establish iSCSI sessions by logging into each target.

   sudo iscsiadm --mode node -l all

   The expected output from the command is:

   
   [ec2-user@ip-172-30-15-58 ~]$ sudo iscsiadm --mode node -l all

4. View and validate a list of active iSCSI sessions.

   sudo iscsiadm --mode session

   Return the iSCSI sessions.
5. Verify that the LUNs were imported into the host.

```
sudo sanlun lun show
```

This will return a list of Oracle LUNs from FSx.
6. Configure the `multipath.conf` file with following default and blacklist entries.
sudo vi /etc/multipath.conf

defaults {
    find_multipaths yes
    user_friendly_names yes
}

blacklist {
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
    devnode "^hd[a-z]"
    devnode "^cciss.*"
}

7. Start the multipath service.

    sudo systemctl start multipathd

Now multipath devices appear in the /dev/mapper directory.

```
[ec2-user@ip-172-30-15-58 ~]$ ls -l /dev/mapper
total 0
lrwxrwxrwx 1 root root 7 Mar 21 20:13 3600a09806c574235472455534e68512d -> ../dm-0
lrwxrwxrwx 1 root root 7 Mar 21 20:13 3600a09806c574235472455534e685141 -> ../dm-1
lrwxrwxrwx 1 root root 7 Mar 21 20:13 3600a09806c574235472455534e685142 -> ../dm-2
lrwxrwxrwx 1 root root 7 Mar 21 20:13 3600a09806c574235472455534e685143 -> ../dm-3
lrwxrwxrwx 1 root root 7 Mar 21 20:13 3600a09806c574235472455534e685144 -> ../dm-4
lrwxrwxrwx 1 root root 7 Mar 21 20:13 3600a09806c574235472455534e685145 -> ../dm-5
lrwxrwxrwx 1 root root 7 Mar 21 18:19 control
```

8. Log into the FSx cluster as the fsxadmin user via SSH to retrieve the serial-hex number for each LUN start with 6c574xxx..., the HEX number start with 3600a0980, which is AWS vendor ID.

    lun show -fields serial-hex
FsxId02ad7bf3476b741df::> lun show -fields serial-hex

<table>
<thead>
<tr>
<th>vserver</th>
<th>path</th>
<th>serial-hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>svm_ora</td>
<td>/vol/ora_01_biny/ora_01_biny_01</td>
<td>6c574235472455534e68512d</td>
</tr>
<tr>
<td>svm_ora</td>
<td>/vol/ora_01_data/ora_01_data_01</td>
<td>6c574235472455534e685141</td>
</tr>
<tr>
<td>svm_ora</td>
<td>/vol/ora_01_data/ora_01_data_02</td>
<td>6c574235472455534e685142</td>
</tr>
<tr>
<td>svm_ora</td>
<td>/vol/ora_01_data/ora_01_data_03</td>
<td>6c574235472455534e685143</td>
</tr>
<tr>
<td>svm_ora</td>
<td>/vol/ora_01_data/ora_01_data_04</td>
<td>6c574235472455534e685144</td>
</tr>
<tr>
<td>svm_ora</td>
<td>/vol/ora_01_logs/ora_01_logs_01</td>
<td>6c574235472455534e685145</td>
</tr>
<tr>
<td>svm_ora</td>
<td>/vol/ora_01_logs/ora_01_logs_02</td>
<td>6c574235472455534e685146</td>
</tr>
</tbody>
</table>

7 entries were displayed.

9. Update the `/dev/multipath.conf` file to add a user-friendly name for the multipath device.

```
sudo vi /etc/multipath.conf
```

with following entries:
multipaths {
multipath {
    wwid 3600a09806c574235472455534e68512d
    alias ora_01_biny_01
}
multipath {
    wwid 3600a09806c574235472455534e685141
    alias ora_01_data_01
}
multipath {
    wwid 3600a09806c574235472455534e685142
    alias ora_01_data_02
}
multipath {
    wwid 3600a09806c574235472455534e685143
    alias ora_01_data_03
}
multipath {
    wwid 3600a09806c574235472455534e685144
    alias ora_01_data_04
}
multipath {
    wwid 3600a09806c574235472455534e685145
    alias ora_01_logs_01
}
multipath {
    wwid 3600a09806c574235472455534e685146
    alias ora_01_logs_02
}
}

10. Reboot the multipath service to verify that the devices under /dev/mapper have changed to LUN names versus serial-hex IDs.

    sudo systemctl restart multipathd

Check /dev/mapper to return as following:
11. Partition the binary LUN with a single primary partition.

```
sudo fdisk /dev/mapper/ora_01_biny_01
```

12. Format the partitioned binary LUN with an XFS file system.

```
sudo mkfs.xfs /dev/mapper/ora_01_biny_01p1
```

13. Mount the binary LUN to /u01.

```
sudo mount -t xfs /dev/mapper/ora_01_biny_01p1 /u01
```

14. Change /u01 mount point ownership to the Oracle user and it's associated primary group.

```
sudo chown oracle:oinstall /u01
```

15. Find the UUI of the binary LUN.

```
sudo blkid /dev/mapper/ora_01_biny_01p1
```

16. Add a mount point to /etc/fstab.
sudo vi /etc/fstab

Add the following line.

```
UUID=d89fb1c9-4f89-4de4-b4d9-17754036d11d /u01 xfs
defaults,nofail 0 2
```

It is important to mount the binary with only the UUID and with the nofail option to avoid possible root-lock issues during EC2-instance reboot.

17. As the root user, add the udev rule for Oracle devices.

```
vi /etc/udev/rules.d/99-oracle-asmdevices.rules
```

Include following entries:

```
ENV{DM_NAME}=="ora*", GROUP="oinstall", OWNER="oracle",
MODE="660"
```

18. As the root user, reload the udev rules.

```
udevadm control --reload-rules
```

19. As the root user, trigger the udev rules.

```
udevadm trigger
```

20. As the root user, reload multipathd.

```
systemctl restart multipathd
```

21. Reboot the EC2 instance host.

---

*Oracle grid infrastructure installation*
1. Log into the EC2 instance as the ec2-user via SSH and enable password authentication by uncommenting `PasswordAuthentication yes` and then commenting out `PasswordAuthentication no`.

```
sudo vi /etc/ssh/sshd_config
```

2. Restart the sshd service.

```
sudo systemctl restart sshd
```

3. Reset the Oracle user password.

```
sudo passwd oracle
```

4. Log in as the Oracle Restart software owner user (oracle). Create an Oracle directory as follows:

```
mkdir -p /u01/app/oracle
mkdir -p /u01/app/oraInventory
```

5. Change the directory permission setting.

```
chmod -R 775 /u01/app
```

6. Create a grid home directory and change to it.

```
mkdir -p /u01/app/oracle/product/19.0.0/grid
cd /u01/app/oracle/product/19.0.0/grid
```

7. Unzip the grid installation files.

```
unzip -q /tmp/archive/LINUX.X64_193000_grid_home.zip
```

8. From grid home, delete the OPatch directory.

```
rm -rf OPatch
```

9. From grid home, unzip `p6880880_190000_Linux-x86-64.zip`.
10. From grid home, revise `cv/admin/cvu_config`, uncomment and replace `CV_ASSUME_DISTID=OEL5` with `CV_ASSUME_DISTID=OL7`.

```
vi cv/admin/cvu_config
```

11. Prepare a `gridsetup.rsp` file for silent installation and place the rsp file in the `/tmp/archive` directory. The rsp file should cover sections A, B, and G with the following information:

```
INVENTORY_LOCATION=/u01/app/oraInventory
oracle.install.option=HA_CONFIG
ORACLE_BASE=/u01/app/oracle
oracle.install.asm.OSDBA=dba
oracle.install.asm.OSOPER=oper
oracle.install.asm.OSASM=asm
oracle.install.asm.SYSASMPassword="SetPWD"
oracle.install.asm.diskGroup.name=DATA
oracle.install.asm.diskGroup.redundancy=EXTERNAL
oracle.install.asm.diskGroup.AUSize=4
oracle.install.asm.diskGroup.disks=/dev/mapper/ora_01_data_01,/dev/mapper/ora_01_data_02,/dev/mapper/ora_01_data_03,/dev/mapper/ora_01_data_04
oracle.install.asm.diskGroup.diskDiscoveryString=/dev/mapper/*
oracle.install.asm.monitorPassword="SetPWD"
oracle.install.asm.configureAFD=true
```

12. Log into the EC2 instance as the root user and set `ORACLE_HOME` and `ORACLE_BASE`.

```
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/grid
export ORACLE_BASE=/tmp
cd /u01/app/oracle/product/19.0.0/grid/bin
```

13. Provision disk devices for use with the Oracle ASM filter driver.
14. Install `cvuqdisk-1.0.10-1.rpm`.

```bash
rpm -ivh /u01/app/oracle/product/19.0.0/grid/cv/rpm/cvuqdisk-1.0.10-1.rpm
```

15. Unset `$ORACLE_BASE`.

```bash
unset ORACLE_BASE
```

16. Log into the EC2 instance as the Oracle user and extract the patch in the `/tmp/archive` folder.

```bash
unzip /tmp/archive/p34762026_190000_Linux-x86-64.zip -d /tmp/archive
```

17. From grid home `/u01/app/oracle/product/19.0.0/grid` and as the oracle user, launch `gridSetup.sh` for grid infrastructure installation.

```bash
./gridSetup.sh -applyRU /tmp/archive/34762026/ -silent
-responseFile /tmp/archive/gridsetup.rsp
```

Ignore the warnings about wrong groups for grid infrastructure. We are using a single Oracle user to manage Oracle Restart, so this is expected.

18. As root user, execute the following script(s):

```bash
/u01/app/oraInventory/orainstRoot.sh
/u01/app/oracle/product/19.0.0/grid/root.sh
```

19. As root user, reload the multipathd.
20. As the Oracle user, execute the following command to complete the configuration:

```
/u01/app/oracle/product/19.0.0/grid/gridSetup.sh -executeConfigTools -responseFile /tmp/archive/gridsetup.rsp -silent
```

21. As the Oracle user, create the LOGS disk group.

```
bin/asmca -silent -sysAsmPassword 'yourPWD' -asmsnmpPassword 'yourPWD' -createDiskGroup -diskGroupName LOGS -disk 'AFD:LOGS*' -redundancy EXTERNAL -au_size 4
```

22. As the Oracle user, validate grid services after installation configuration.

```
bin/crsctl stat res -t
+ Name                Target  State        Server
  State details
  Local Resources
  ora.DATA.dg         ONLINE  ONLINE       ip-172-30-15-58
  STABLE
  ora.LISTENER.lsnr   ONLINE  ONLINE       ip-172-30-15-58
  STABLE
  ora.LOGS.dg         ONLINE  ONLINE       ip-172-30-15-58
  STABLE
  ora.asm             ONLINE  ONLINE       ip-172-30-15-58
  Started,STABLE
  ora-ons             OFFLINE OFFLINE      ip-172-30-15-58
  STABLE
  Cluster Resources
  ora.cssd            ONLINE  ONLINE       ip-172-30-15-58
  STABLE
  ora.diskmon         OFFLINE OFFLINE      STABLE
  ora.driver.afd      ONLINE  ONLINE       ip-172-30-15-58
  STABLE
  ora.evmd            ONLINE  ONLINE       ip-172-30-15-58
  STABLE
```

23. Validate ASM filter driver status.
Oracle database installation

```
[oracle@ip-172-30-15-58 grid]$ export ORACLE_HOME=/u01/app/oracle/product/19.0.0/grid
[oracle@ip-172-30-15-58 grid]$ export ORACLE_SID=+ASM
[oracle@ip-172-30-15-58 grid]$ export PATH=$PATH:$ORACLE_HOME/bin
[oracle@ip-172-30-15-58 grid]$ asmcmd
ASMCMD> lsdg
State Type Rebal Sector Logical_Sector Block AU Total_MB Free_MB Req_mir_free_MB Usable_file_MB Offline_disks Voting_files Name
MOUNTED EXTERN N 512 512 4096 1048576
81920 81847 0 81847 0 N DATA/
MOUNTED EXTERN N 512 512 4096 1048576
81920 81853 0 81853 0 N LOGS/
ASMCMD> afd_state
ASMCMD-9526: The AFD state is 'LOADED' and filtering is 'ENABLED' on host 'ip-172-30-15-58.ec2.internal'
```
1. Log in as the Oracle user and unset $ORACLE_HOME and $ORACLE_SID if it is set.

```bash
unset ORACLE_HOME
unset ORACLE_SID
```

2. Create the Oracle DB home directory and change to it.

```bash
mkdir /u01/app/oracle/product/19.0.0/db1
cd /u01/app/oracle/product/19.0.0/db1
```

3. Unzip the Oracle DB installation files.

```bash
unzip -q /tmp/archive/LINUX.X64_193000_db_home.zip
```

4. From the DB home, delete the OPatch directory.

```bash
rm -rf OPatch
```

5. From DB home, unzip p6880880_190000_Linux-x86-64.zip.

```bash
unzip -q /tmp/archive/p6880880_190000_Linux-x86-64.zip
```

6. From DB home, revise cv/admin/cvu_config, and uncomment and replace
   CV_ASSUME_DISTID=OEL5 with CV_ASSUME_DISTID=OL7.

```bash
vi cv/admin/cvu_config
```

7. From the /tmp/archive directory, unpack the DB 19.18 RU patch.

```bash
unzip p34765931_190000_Linux-x86-64.zip
```

8. Prepare the DB silent install rsp file in /tmp/archive/dbinstall.rsp directory with the following values:
9. From db1 home /u01/app/oracle/product/19.0.0/db1, execute silent software-only DB installation.

```
./runInstaller -applyRU /tmp/archive/34765931/ -silent
-ignorePrereqFailure -responseFile /tmp/archive/dbinstall.rsp
```

10. As root user, run the `root.sh` script after software-only installation.

```
/u01/app/oracle/product/19.0.0/db1/root.sh
```

11. As Oracle user, create the `dbca.rsp` file with the following entries:
gdbName=db1.demo.netapp.com
sid=db1
createAsContainerDatabase=true
numberOfPDBs=3
pdbName=db1_pdb
useLocalUndoForPDBs=true
pdbAdminPassword="yourPWD"
templateName=General_Purpose.dbc
sysPassword="yourPWD"
systemPassword="yourPWD"
dbsnmpPassword="yourPWD"
datafileDestination=+DATA
recoveryAreaDestination=+LOGS
storageType=ASM
diskGroupName=DATA
characterSet=AL32UTF8
nationalCharacterSet=AL16UTF16
listeners=LISTENER
databaseType=MULTIPURPOSE
automaticMemoryManagement=false
totalMemory=8192

12. As Oracle user, launch DB creation with dbca.
bin/dbca -silent -createDatabase -responseFile /tmp/archive/dbca.rsp

output:
Prepare for db operation
7% complete
Registering database with Oracle Restart
11% complete
Copying database files
33% complete
Creating and starting Oracle instance
35% complete
38% complete
42% complete
45% complete
48% complete
Completing Database Creation
53% complete
55% complete
56% complete
Creating Pluggable Databases
60% complete
64% complete
69% complete
78% complete
Executing Post Configuration Actions
100% complete
Database creation complete. For details check the logfiles at:
   /u01/app/oracle/cfgtoollogs/dbca/db1.
Database Information:
Global Database Name:db1.demo.netapp.com
System Identifier(SID):db1
Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/db1/db1.log"
for further details.

13. As Oracle user, validate Oracle Restart HA services after DB creation.
Local Resources

<table>
<thead>
<tr>
<th>Name</th>
<th>Target</th>
<th>State</th>
<th>Server</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora.DATA.dg</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.LISTENER.lsnr</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.LOGS.dg</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.asm</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>Started,STABLE</td>
</tr>
<tr>
<td>ora.ons</td>
<td>OFFLINE</td>
<td>OFFLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
</tbody>
</table>

Cluster Resources

<table>
<thead>
<tr>
<th>Name</th>
<th>Target</th>
<th>State</th>
<th>Server</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora.cssd</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.db1.db</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>Open,HOME=/u01/app/oracle/product/19.0.0/db1,STABLE</td>
</tr>
<tr>
<td>ora.diskmon</td>
<td>OFFLINE</td>
<td>OFFLINE</td>
<td></td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.driver.afd</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.evmd</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
</tbody>
</table>

14. Set the Oracle user .bash_profile.

```
vi ~/.bash_profile
```

15. Add following entries:

```
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db1
export ORACLE_SID=db1
export PATH=$PATH:$ORACLE_HOME/bin
alias asm='export ORACLE_HOME=/u01/app/oracle/product/19.0.0/grid;export ORACLE_SID=+ASM;export PATH=$PATH:$ORACLE_HOME/bin'
```

16. Validate the CDB/PDB created.

```
/home/oracle/.bash_profile
sqlplus / as sysdba
```
SQL> select name, open_mode from v$database;

NAME   OPEN_MODE
DB1    READ WRITE

SQL> select name from v$datafile;

NAME
+DATA/DB1/DATAFILE/system.256.1132176177
+DATA/DB1/DATAFILE/sysaux.257.1132176221
+DATA/DB1/DATAFILE/undotbs1.258.1132176247
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/system.265.1132177009
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/sysaux.266.1132177009
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/undotbs1.267.1132177009
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/system.271.1132177853
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/sysaux.272.1132177853
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/undotbs1.270.1132177853
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/users.274.1132177871

NAME
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/system.276.1132177871
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/sysaux.277.1132177871
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/undotbs1.275.1132177871
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/users.279.1132177889
+DATA/DB1/F78529A14DD8BB18E0533A0F1EACB8ED/DATAFILE/system.281.1132177889
+DATA/DB1/F78529A14DD8BB18E0533A0F1EACB8ED/DATAFILE/sysaux.282.1132177889
+DATA/DB1/F78529A14DD8BB18E0533A0F1EACB8ED/DATAFILE/undotbs1.280.1132177889
19 rows selected.

SQL> show pdbs

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>CON_NAME</th>
<th>OPEN MODE</th>
<th>RESTRICTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>READ ONLY</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>DB1_PDB1</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>4</td>
<td>DB1_PDB2</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>5</td>
<td>DB1_PDB3</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
</tbody>
</table>

17. Set the DB recovery destination size to the +LOGS disk group size.

```
alter system set db_recovery_file_dest_size = 80G scope=both;
```

18. Log into the database with sqlplus and enable archive log mode.

```
sqlplus /as sysdba.
shutdown immediate;
startup mount;
alter database archivelog;
alter database open;
```

This completes Oracle 19c version 19.18 Restart deployment on an Amazon FSx for ONTAP and EC2 compute instance. If desired, NetApp recommends relocating the Oracle control file and online log files to the +LOGS disk group.

Automated deployment option

NetApp will release a fully automated solution deployment toolkit with Ansible to facilitate the implementation of this solution. Please check back for the availability of the toolkit. After it is released, a link will be posted here.

Oracle Database backup, restore, and clone with SnapCenter Service

See SnapCenter Services for Oracle for details on Oracle database backup, restore, and clone with NetApp BlueXP console.
Where to find additional information

To learn more about the information described in this document, review the following documents and/or websites:

- Installing Oracle Grid Infrastructure for a Standalone Server with a New Database Installation

- Installing and Configuring Oracle Database Using Response Files

- Amazon FSx for NetApp ONTAP
  https://aws.amazon.com/fsx/netapp-ontap/

- Amazon EC2
  https://aws.amazon.com/ec2/?trk=36c6da98-7b20-48fa-8225-4784bced9843&sc_channel=p&s_kwcid=AL!4422!3!467723097970le!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARlsAHzSrdqwQrghn6l71jiWzSeaT9Uh1-vY-VfihJixF-xnv5rWvn2S7RqZOTQ0aAh7eEALw_wcb:G:s&s_kwcid=AL!4422!3!467723097970le!!g!!aws%20ec2

Oracle Database Deployment on AWS EC2 and FSx Best Practices

WP-7357: Oracle Database Deployment on EC2 and FSx Best Practices Introduction

Allen Cao, Niyaz Mohamed, Jeffrey Steiner, NetApp

Many mission-critical enterprise Oracle databases are still hosted on-premises, and many enterprises are looking to migrate these Oracle databases to a public cloud. Often, these Oracle databases are application centric and thus require user-specific configurations, a capability that is missing from many database-as-a-service public-cloud offerings. Therefore, the current database landscape calls for a public-cloud-based Oracle database solution built from a high-performance, scalable compute and storage service that can accommodate unique requirements. AWS EC2 compute instances and the AWS FSx storage service might be the missing pieces of this puzzle that you can leverage to build and migrate your mission critical Oracle database workloads to a public cloud.

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for enterprises. The simple Amazon EC2 web-service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon’s proven computing environment.

Amazon FSx for ONTAP is an AWS storage service that uses industry-leading NetApp ONTAP block and file storage, which exposes NFS, SMB, and iSCSI. With such a powerful storage engine, it has never been easier to relocate mission-critical Oracle database apps to AWS with sub-millisecond response times, multiple GBps
of throughput, and 100,000+ IOPS per database instance. Better yet, the FSx storage service comes with native replication capability that allows you to easily migrate your on-premises Oracle database to AWS or to replicate your mission critical Oracle database to a secondary AWS availability zone for HA or DR.

The goal of this documentation is to provide step-by-step processes, procedures, and best-practice guidance on how to deploy and configure an Oracle database with FSx storage and an EC2 instance that delivers performance similar to an on-premises system. NetApp also provides an automation toolkit that automates most of the tasks that are required for the deployment, configuration, and management of your Oracle database workload in the AWS public cloud.

To learn more about the solution and use case, take a look at following overview video:

Modernize your Oracle database with hybrid cloud in AWS and FSx ONTAP, Part1 - Use case and solution architecture

Solution architecture

The following architecture diagram illustrates a highly available Oracle database deployment on an AWS EC2 instance with the FSx storage service. A similar deployment scheme but with the standby in a different region can be set up for disaster recovery.

Within the environment, the Oracle compute instance is deployed via an AWS EC2 instance console. There are multiple EC2 instance types available from the console. NetApp recommends deploying a database-oriented EC2 instance type such as an m5 Ami image with RedHat enterprise Linux 8 and up to 10Gps of network bandwidth.

Oracle database storage on FSx volumes on the other hand is deployed with the AWS FSx console or CLI. The Oracle binary, data, or log volumes are subsequently presented and mounted on an EC2 instance Linux host. Each data or log volume can have multiple LUNs allocated depending on the underlying storage protocol employed.
An FSx storage cluster is designed with double redundancy, so that both the primary and standby storage clusters are deployed in two different availability zones. Database volumes are replicated from a primary FSx cluster to a standby FSx cluster at a user-configurable interval for all Oracle binary, data, and log volumes.

This high availability Oracle environment is managed with an Ansible controller node and a SnapCenter backup server and UI tool. Oracle installation, configuration, and replication are automated using Ansible playbook-based tool kits. Any update to the Oracle EC2 instance kernel operating system or Oracle patching can be executed in parallel to keep the primary and standby in sync. In fact, the initial automation setup can be easily expanded to perform some repeating daily Oracle tasks if needed.

SnapCenter provides workflows for Oracle database point-in-time recovery or for database cloning at either the primary or standby zones if needed. Through the SnapCenter UI, you can configure Oracle database backup and replication to standby FSx storage for high availability or disaster recovery based on your RTO or RPO objectives.

The solution provides an alternative process that delivers capabilities similar to those available from Oracle RAC and Data Guard deployment.

**Factors to consider for Oracle database deployment**

A public cloud provides many choices for compute and storage, and using the correct type of compute instance and storage engine is a good place to start for database deployment. You should also select compute and storage configurations that are optimized for Oracle databases.

The following sections describe the key considerations when deploying Oracle database in an AWS public cloud on an EC2 instance with FSx storage.

**VM performance**

Selecting the right VM size is important for optimal performance of a relational database in a public cloud. For better performance, NetApp recommends using an EC2 M5 Series instance for Oracle deployment, which is optimized for database workloads. The same instance type is also used to power a RDS instance for Oracle by AWS.

- Choose the correct vCPU and RAM combination based on workload characteristics.
- Add swap space to a VM. The default EC2 instance deployment does not create a swap space, which is not optimal for a database.

**Storage layout and settings**

NetApp recommends the following storage layout:

- For NFS storage, the recommended volume layout is three volumes: one for the Oracle binary; one for Oracle data and a duplicate control file; and one for the Oracle active log, archived log, and control file.
For iSCSI storage, the recommended volume layout is three volumes: one for the Oracle binary; one for Oracle data and a duplicate control file; and one for the Oracle active log, archived log, and control file. However, each data and log volume ideally should contain four LUNs. The LUNs are ideally balanced on the HA cluster nodes.

For storage IOPS and throughput, you can choose the threshold for provisioned IOPS and throughput for the FSx storage cluster, and these parameters can be adjusted on the fly anytime the workload changes.

- The auto IOPS setting is three IOPS per GiB of allocated storage capacity or user defined storage up to 80,000.
- The throughput level is incremented as follow: 128, 256, 512, 1024, 2045 MBps.

Review the Amazon FSx for NetApp ONTAP performance documentation when sizing throughput and IOPS.

NFS configuration

Linux, the most common operating system, includes native NFS capabilities. Oracle offers the direct NFS (dNFS) client natively integrated into Oracle. Oracle has supported NFSv3 for over 20 years. dNFS is supported with NFSv3 with all versions of Oracle. NFSv4 is supported with all OS’s that follow the NFSv4 standard. dNFS support for NFSv4 requires Oracle 12.1.0.2 or higher. NFSv4.1 requires specific OS support. Consult the NetApp Interoperability Matrix Tool (IMT) for supported OS’s. dNFS support for NFSv4.1 requires Oracle version 19.3.0.0 or higher.

Automated Oracle deployment using the NetApp automation toolkit automatically configures dNFS on NFSv3.

Other factors to consider:

- TCP slot tables are the NFS equivalent of host-bus-adapter (HBA) queue depth. These tables control the number of NFS operations that can be outstanding at any one time. The default value is usually 16, which
is far too low for optimum performance. The opposite problem occurs on newer Linux kernels, which can automatically increase the TCP slot table limit to a level that saturates the NFS server with requests.

For optimum performance and to prevent performance problems, adjust the kernel parameters that control the TCP slot tables to 128.

```
sysctl -a | grep tcp.*.slot_table
```

• The following table provides recommended NFS mount options for Linux NFSv3 - single instance.

<table>
<thead>
<tr>
<th>File Type</th>
<th>Mount Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control files</td>
<td>rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536</td>
</tr>
<tr>
<td>Data files</td>
<td></td>
</tr>
<tr>
<td>Redo logs</td>
<td></td>
</tr>
<tr>
<td>ORACLE_HOME</td>
<td>rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536</td>
</tr>
<tr>
<td>ORACLE_BASE</td>
<td></td>
</tr>
</tbody>
</table>

Before using dNFS, verify that the patches described in Oracle Doc 1495104.1 are installed. The NetApp Support matrix for NFSv3 and NFSv4 do not include specific operating systems. All OSs that obey the RFC are supported. When searching the online IMT for NFSv3 or NFSv4 support, do not select a specific OS because no matches will be displayed. All OSs are implicitly supported by the general policy.

High availability

As indicated in the solution architecture, HA is built on storage-level replication. Therefore, the startup and availability of Oracle is contingent on how quickly the compute and storage can be brought up and recovered. See the following key factors:

• Have a standby compute instance ready and synced up with the primary through Ansible parallel update to both hosts.
• Replicate the binary volume from the primary for standby purposes so that you do not need to install Oracle at the last minute and figure out what needs to be installed and patched.
• Replication frequency dictates how fast the Oracle database can be recovered to make service available. There is a trade off between the replication frequency and storage consumption.
• Leverage automation to make recovery and switch over to standby quick and free of human error. NetApp provides an automation toolkit for this purpose.

Step-by-Step Oracle Deployment Procedures on AWS EC2 and FSx

This section describes the deployment procedures of deploying Oracle RDS custom database with FSx storage.

Deploy an EC2 Linux instance for Oracle via EC2 console

If you are new to AWS, you first need to set up an AWS environment. The documentation tab at the AWS website landing page provides EC2 instruction links on how to deploy a Linux EC2 instance that can be used to host your Oracle database via the AWS EC2 console. The following section is a summary of these steps.
For details, see the linked AWS EC2-specific documentation.

Setting up your AWS EC2 environment

You must create an AWS account to provision the necessary resources to run your Oracle environment on the EC2 and FSx service. The following AWS documentation provides the necessary details:

- Set up to use Amazon EC2

Key topics:

- Sign up for AWS.
- Create a key pair.
- Create a security group.

Enabling multiple availability zones in AWS account attributes

For an Oracle high availability configuration as demonstrated in the architecture diagram, you must enable at least four availability zones in a region. The multiple availability zones can also be situated in different regions to meet the required distances for disaster recovery.

Creating and connecting to an EC2 instance for hosting Oracle database

See the tutorial Get started with Amazon EC2 Linux instances for step-by-step deployment procedures and best practices.

Key topics:

- Overview.
- Prerequisites.
• Step 1: Launch an instance.
• Step 2: Connect to your instance.
• Step 3: Clean up your instance.

The following screen shots demonstrate the deployment of an m5-type Linux instance with the EC2 console for running Oracle.

1. From the EC2 dashboard, click the yellow Launch Instance button to start the EC2 instance deployment workflow.

2. In Step 1, select "Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0b0af3577fe5e3532 (64-bit x86) / ami-01fc429821bf1f4b4 (64-bit Arm)."

3. In Step 2, select an m5 instance type with the appropriate CPU and memory allocation based on your Oracle database workload. Click "Next: Configure Instance Details."
4. In Step 3, choose the VPC and subnet where the instance should be placed and enable public IP assignment. Click "Next: Add Storage."

5. In Step 4, allocate enough space for the root disk. You may need the space to add a swap. By default, EC2 instance assign zero swap space, which is not optimal for running Oracle.
6. In Step 5, add a tag for instance identification if needed.

7. In Step 6, select an existing security group or create a new one with the desired inbound and outbound policy for the instance.
8. In Step 7, review the instance configuration summary, and click Launch to start instance deployment. You are prompted to create a key pair or select a key pair for access to the instance.
9. Log into EC2 instance using an SSH key pair. Make changes to your key name and instance IP address as appropriate.

ssh -i ora-db1v2.pem ec2-user@54.80.114.77

You need to create two EC2 instances as primary and standby Oracle servers in their designated availability zone as demonstrated in the architecture diagram.

**Provision FSx for ONTAP file systems for Oracle database storage**

EC2 instance deployment allocates an EBS root volume for the OS. FSx for ONTAP file systems provides Oracle database storage volumes, including the Oracle binary, data, and log volumes. The FSx storage NFS volumes can be either provisioned from the AWS FSx console or from Oracle installation, and configuration automation that allocates the volumes as the user configures in a automation parameter file.

**Creating FSx for ONTAP file systems**

Referred to this documentation [Managing FSx for ONTAP file systems](#) for creating FSx for ONTAP file systems.

Key considerations:

- **SSD storage capacity.** Minimum 1024 GiB, maximum 192 TiB.
- **Provisioned SSD IOPS.** Based on workload requirements, a maximum of 80,000 SSD IOPS per file system.
- **Throughput capacity.**
• Set administrator fsxadmin/vsadmin password. Required for FSx configuration automation.

• Backup and maintenance. Disable automatic daily backups; database storage backup is executed through SnapCenter scheduling.

• Retrieve the SVM management IP address as well as protocol-specific access addresses from SVM details page. Required for FSx configuration automation.

See the following step-by-step procedures for setting up either a primary or standby HA FSx cluster.

1. From the FSx console, click Create File System to start the FSx provision workflow.

2. Select Amazon FSx for NetApp ONTAP. Then click Next.
3. Select Standard Create and, in File System Details, name your file system, Multi-AZ HA. Based on your database workload, choose either Automatic or User-Provisioned IOPS up to 80,000 SSD IOPS. FSx storage comes with up to 2TiB NVMe caching at the backend that can deliver even higher measured IOPS.
4. In the Network & Security section, select the VPC, security group, and subnets. These should be created before FSx deployment. Based on the role of the FSx cluster (primary or standby), place the FSx storage nodes in the appropriate zones.
5. In the Security & Encryption section, accept the default, and enter the fsxadmin password.
6. Enter the SVM name and the vsadmin password.

![Default storage virtual machine configuration]

**Storage virtual machine name**

fsxora_prod

**SVM administrative password**

Password for this SVM’s "vsadmin" user, which you can use to access the ONTAP CLI or REST API.

- [ ] Don’t specify a password
- [x] Specify a password

**Password**

**Confirm password**

**Active Directory**

Joining an Active Directory enables access from Windows and MacOS clients over the SMB protocol.

- [x] Do not join an Active Directory
- [ ] Join an Active Directory

7. Leave the volume configuration blank; you do not need to create a volume at this point.
8. Review the Summary page, and click Create File System to complete FSx file system provision.
Provisioning of database volumes for Oracle database

See [Managing FSx for ONTAP volumes - creating a volume](#) for details.

Key considerations:

- Sizing the database volumes appropriately.
- Disabling capacity pool tiering policy for performance configuration.
- Enabling Oracle dNFS for NFS storage volumes.
- Setting up multipath for iSCSI storage volumes.

Create database volume from FSx console

From the AWS FSx console, you can create three volumes for Oracle database file storage: one for the Oracle binary, one for the Oracle data, and one for the Oracle log. Make sure that volume naming matches the Oracle host name (defined in the hosts file in the automation toolkit) for proper identification. In this example, we use db1 as the EC2 Oracle host name instead of a typical IP-address-based host name for an EC2 instance.
Create volume

File system
ONTAP | fs-0a51a3f08922224d5 | rdscustomfs007

Storage virtual machine
svm-005c6edf027866ca4 | fsx

Volume name
db1_bin

Maximum of 203 alphanumeric characters, plus _.

Junction path
/db1_bin

The location within your file system where your volume will be mounted.

Volume size
51200

Minimum 20 MiB; Maximum 104857600 MiB

Storage efficiency
Select whether you would like to enable ONTAP storage efficiencies on your volume: deduplication, compression, and compaction.

- Enabled (recommended)
- Disabled

Capacity pool tiering policy
You can optionally enable automatic tiering of your data to lower-cost capacity pool storage.

- None
Create volume

**File system**

ONTAP | fs-0a51a3f08922224d5 | rdscustomfs007

**Storage virtual machine**

svm-005c6edf027866ca4 | fsx

**Volume name**

db1_data

Maximum of 203 alphanumeric characters, plus _.

**Junction path**

/db1_data

The location within your file system where your volume will be mounted.

**Volume size**

512000

Minimum 20 MiB; Maximum 104857600 MiB

**Storage efficiency**

Select whether you would like to enable ONTAP storage efficiencies on your volume: deduplication, compression, and compaction.

- **Enabled (recommended)**
- **Disabled**

**Capacity pool tiering policy**

You can optionally enable automatic tiering of your data to lower-cost capacity pool storage.

None
Creating iSCSI LUNs is not currently supported by the FSx console. For iSCSI LUNs deployment for Oracle, the volumes and LUNs can be created by using automation for ONTAP with the NetApp Automation Toolkit.

Install and configure Oracle on an EC2 instance with FSx database volumes

The NetApp automation team provide an automation kit to run Oracle installation and configuration on EC2 instances according to best practices. The current version of the automation kit supports Oracle 19c on NFS with the default RU patch 19.8. The automation kit can be easily adapted for other RU patches if needed.
Prepare a Ansible controller to run automation

Follow the instruction in the section "Creating and connecting to an EC2 instance for hosting Oracle database" to provision a small EC2 Linux instance to run the Ansible controller. Rather than using RedHat, Amazon Linux t2.large with 2vCPU and 8G RAM should be sufficient.

Retrieve NetApp Oracle deployment automation toolkit

Log into the EC2 Ansible controller instance provisioned from step 1 as ec2-user and from the ec2-user home directory, execute the git clone command to clone a copy of the automation code.

```bash
git clone https://github.com/NetApp-Automation/na_oracle19c_deploy.git
```

```bash
git clone https://github.com/NetApp-Automation/na_rds_fsx_oranfs_config.git
```

Execute automated Oracle 19c deployment using automation toolkit

See these detailed instruction CLI deployment Oracle 19c Database to deploy Oracle 19c with CLI automation. There is a small change in command syntax for playbook execution because you are using an SSH key pair instead of a password for host access authentication. The following list is a high level summary:

1. By default, an EC2 instance uses an SSH key pair for access authentication. From Ansible controller automation root directories /home/ec2-user/na_oracle19c_deploy, and /home/ec2-user/na_rds_fsx_oranfs_config, make a copy of the SSH key accesststkey.pem for the Oracle host deployed in the step "Creating and connecting to an EC2 instance for hosting Oracle database."

2. Log into the EC2 instance DB host as ec2-user, and install the python3 library.

```bash
sudo yum install python3
```

3. Create a 16G swap space from the root disk drive. By default, an EC2 instance creates zero swap space. Follow this AWS documentation: How do I allocate memory to work as swap space in an Amazon EC2 instance by using a swap file?.

4. Return to the Ansible controller (cd /home/ec2-user/na_rds_fsx_oranfs_config), and execute the preclone playbook with the appropriate requirements and linux_config tags.

```bash
ansible-playbook -i hosts rds_preclone_config.yml -u ec2-user --private -key accesststkey.pem -e @vars/fsx_vars.yml -t requirements_config
```

```bash
ansible-playbook -i hosts rds_preclone_config.yml -u ec2-user --private -key accesststkey.pem -e @vars/fsx_vars.yml -t linux_config
```

5. Switch to the /home/ec2-user/na_oracle19c_deploy-master directory, read the README file, and populate the global vars.yml file with the relevant global parameters.
6. Populate the `host_name.yml` file with the relevant parameters in the `host_vars` directory.

7. Execute the playbook for Linux, and press Enter when prompted for the vsadmin password.

   ```bash
   ansible-playbook -i hosts all_playbook.yml -u ec2-user --private-key accesststkey.pem -t linux_config -e @vars/vars.yml
   ```

8. Execute the playbook for Oracle, and press enter when prompted for the vsadmin password.

   ```bash
   ansible-playbook -i hosts all_playbook.yml -u ec2-user --private-key accesststkey.pem -t oracle_config -e @vars/vars.yml
   ```

   Change the permission bit on the SSH key file to 400 if needed. Change the Oracle host (ansible_host in the `host_vars` file) IP address to your EC2 instance public address.

**Setting up SnapMirror between primary and standby FSx HA cluster**

For high availability and disaster recovery, you can set up SnapMirror replication between the primary and standby FSx storage cluster. Unlike other cloud storage services, FSx enables a user to control and manage storage replication at a desired frequency and replication throughput. It also enables users to test HA/DR without any effect on availability.

The following steps show how to set up replication between a primary and standby FSx storage cluster.

1. Setup primary and standby cluster peering. Log into the primary cluster as the fsxadmin user and execute the following command. This reciprocal create process executes the create command on both the primary cluster and the standby cluster. Replace `standby_cluster_name` with the appropriate name for your environment.

   ```bash
   cluster peer create -peer-addrs standby_cluster_name,inter_cluster_ip_address -username fsxadmin -initial-allowed-vserver-peers *
   ```

2. Set up vServer peering between the primary and standby cluster. Log into the primary cluster as the vsadmin user and execute the following command. Replace `primary_vserver_name`, `standby_vserver_name`, `standby_cluster_name` with the appropriate names for your environment.

   ```bash
   vserver peer create -vserver primary_vserver_name -peer-vserver standby_vserver_name -peer-cluster standby_cluster_name -applications snapmirror
   ```

3. Verify that the cluster and vserver peerings are set up correctly.
4. Create target NFS volumes at the standby FSx cluster for each source volume at the primary FSx cluster. Replace the volume name as appropriate for your environment.

```
vol create -volume dr_db1_bin -aggregate aggr1 -size 50G -state online -policy default -type DP
```

```
vol create -volume dr_db1_data -aggregate aggr1 -size 500G -state online -policy default -type DP
```

```
vol create -volume dr_db1_log -aggregate aggr1 -size 250G -state online -policy default -type DP
```

5. You can also create iSCSI volumes and LUNs for the Oracle binary, Oracle data, and the Oracle log if the iSCSI protocol is employed for data access. Leave approximately 10% free space in the volumes for snapshots.

```
vol create -volume dr_db1_bin -aggregate aggr1 -size 50G -state online -policy default -unix-permissions ---rwxr-xr-x -type RW
```

```
lun create -path /vol/dr_db1_bin/dr_db1_bin_01 -size 45G -ostype linux
```

```
vol create -volume dr_db1_data -aggregate aggr1 -size 500G -state online -policy default -unix-permissions ---rwxr-xr-x -type RW
```

```
lun create -path /vol/dr_db1_data/dr_db1_data_01 -size 100G -ostype linux
```
lun create -path /vol/dr_db1_data/dr_db1_data_02 -size 100G -ostype linux
lun create -path /vol/dr_db1_data/dr_db1_data_03 -size 100G -ostype linux
lun create -path /vol/dr_db1_data/dr_db1_data_04 -size 100G -ostype linux

vol create -volume dr_db1_log -aggregate aggr1 -size 250G -state online -policy default -unix-permissions ---rwxr-xr-x -type RW

lun create -path /vol/dr_db1_log/dr_db1_log_01 -size 45G -ostype linux
lun create -path /vol/dr_db1_log/dr_db1_log_02 -size 45G -ostype linux
lun create -path /vol/dr_db1_log/dr_db1_log_03 -size 45G -ostype linux
lun create -path /vol/dr_db1_log/dr_db1_log_04 -size 45G -ostype linux

6. For iSCSI LUNs, create mapping for the Oracle host initiator for each LUN, using the binary LUN as an example. Replace the igroup with an appropriate name for your environment, and increment the lun-id for each additional LUN.

lun mapping create -path /vol/dr_db1_bin/dr_db1_bin_01 -igroup ip-10-0-1-136 -lun-id 0
lun mapping create -path /vol/dr_db1_data/dr_db1_data_01 -igroup ip-10-0-1-136 -lun-id 1

7. Create a SnapMirror relationship between the primary and standby database volumes. Replace the appropriate SVM name for your environment.
This SnapMirror setup can be automated with a NetApp Automation Toolkit for NFS database volumes. The toolkit is available for download from the NetApp public GitHub site.

```bash
git clone https://github.com/NetApp-Automation/na_ora_hadr_failover_resync.git
```

Read the README instructions carefully before attempting setup and failover testing.

> Replicating the Oracle binary from the primary to a standby cluster might have Oracle license implications. Contact your Oracle license representative for clarification. The alternative is to have Oracle installed and configured at the time of recovery and failover.

SnapCenter Deployment

SnapCenter installation

Follow [Installing the SnapCenter Server](#) to install SnapCenter server. This documentation covers how to install a standalone SnapCenter server. A SaaS version of SnapCenter is in beta review and could be available shortly. Check with your NetApp representative for availability if needed.

Configure SnapCenter plugin for EC2 Oracle host

1. After automated SnapCenter installation, log into SnapCenter as an administrative user for the Window host on which the SnapCenter server is installed.
2. From the left-side menu, click Settings, and then Credential and New to add ec2-user credentials for SnapCenter plugin installation.

3. Reset the ec2-user password and enable password SSH authentication by editing the /etc/ssh/sshd_config file on the EC2 instance host.

4. Verify that the "Use sudo privileges" checkbox is selected. You just reset the ec2-user password in the previous step.
5. Add the SnapCenter server name and the IP address to the EC2 instance host file for name resolution.

```
[ec2-user@ip-10-0-0-151 ~]$ sudo vi /etc/hosts
[ec2-user@ip-10-0-0-151 ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
10.0.1.233  rdscustomvalsc.rdscustomval.com rdscustomvalsc
```

6. On the SnapCenter server Windows host, add the EC2 instance host IP address to the Windows host file C:\Windows\System32\drivers\etc\hosts.

```
10.0.0.151      ip-10-0-0-151.ec2.internal
```

7. In the left-side menu, select Hosts > Managed Hosts, and then click Add to add the EC2 instance host to SnapCenter.
Check Oracle Database, and, before you submit, click More Options.

Check Skip Preinstall Checks. Confirm Skipping Preinstall Checks, and then click Submit After Save.
You are prompted with Confirm Fingerprint, and then click Confirm and Submit.

After successful plugin configuration, the managed host's overall status show as Running.

Configure backup policy for Oracle database

Refer to this section Setup database backup policy in SnapCenter for details on configuring the Oracle database backup policy.

Generally you need create a policy for the full snapshot Oracle database backup and a policy for the Oracle archive-log-only snapshot backup.
You can enable Oracle archive log pruning in the backup policy to control log-archive space. Check "Update SnapMirror after creating a local Snapshot copy" in "Select secondary replication option" as you need to replicate to a standby location for HA or DR.

Configure Oracle database backup and scheduling

Database backup in SnapCenter is user configurable and can be set up either individually or as a group in a resource group. The backup interval depends on the RTO and RPO objectives. NetApp recommends that you run a full database backup every few hours and archive the log backup at a higher frequency such as 10-15 mins for quick recovery.

Refer to the Oracle section of Implement backup policy to protect database for a detailed step-by-step processes for implementing the backup policy created in the section Configure backup policy for Oracle database and for backup job scheduling.

The following image provides an example of the resources groups that are set up to back up an Oracle database.

EC2 and FSx Oracle database management

In addition to the AWS EC2 and FSx management console, the Ansible control node and the SnapCenter UI tool are deployed for database management in this Oracle environment.

An Ansible control node can be used to manage Oracle environment configuration, with parallel updates that keep primary and standby instances in sync for kernel or patch updates. Failover, resync, and failback can be automated with the NetApp Automation Toolkit to archive fast application recovery and availability with Ansible. Some repeatable database management tasks can be executed using a playbook to reduce human errors.

The SnapCenter UI tool can perform database snapshot backup, point-in-time recovery, database cloning, and so on with the SnapCenter plugin for Oracle databases. For more information about Oracle plugin features, see the SnapCenter Plug-in for Oracle Database overview.

The following sections provide details on how key functions of Oracle database management are fulfilled with the SnapCenter UI:

- Database snapshot backups
- Database point-in-time restore
- Database clone creation

Database cloning creates a replica of a primary database on a separate EC2 host for data recovery in the event of logical data error or corruption, and clones can also be used for application testing, debugging, patch validation, and so on.
Taking a snapshot

An EC2/FSx Oracle database is regularly backed up at intervals configured by the user. A user can also take a one-off snapshot backup at any time. This applies to both full-database snapshot backups as well as archive-log-only snapshot backups.

Taking a full database snapshot

A full database snapshot includes all Oracle files, including data files, control files, and archive log files.

1. Log into the SnapCenter UI and click Resources in the left-side menu. From the View dropdown, change to the Resource Group view.

2. Click the full backup resource name, and then click the Backup Now icon to initiate an add-hoc backup.

3. Click Backup and then confirm the backup to start a full database backup.
From the Resource view for the database, open the database Managed Backup Copies page to verify that the one-off backup completed successfully. A full database backup creates two snapshots: one for the data volume and one for the log volume.

Taking an archive log snapshot

An archive log snapshot is only taken for the Oracle archive log volume.

1. Log into the SnapCenter UI and click the Resources tab in the left-side menu bar. From the View dropdown, change to the Resource Group view.
2. Click the log backup resource name, and then click the Backup Now icon to initiate an add-hoc backup for archive logs.

3. Click Backup and then confirm the backup to start an archive log backup.
From the Resource view for the database, open the database Managed Backup Copies page to verify that the one-off archive log backup completed successfully. An archive log backup creates one snapshot for the log volume.

Restoring to a point in time

SnapCenter-based restore to a point in time is executed on the same EC2 instance host. Complete the following steps to perform the restore:

1. From the SnapCenter Resources tab > Database view, click the database name to open the database backup.

2. Select the database backup copy and the desired point in time to be restored. Also mark down the corresponding SCN number for the point in time. The point-in-time restore can be performed using either the time or the SCN.
3. Highlight the log volume snapshot and click the Mount button to mount the volume.

4. Choose the primary EC2 instance to mount the log volume.
5. Verify that the mount job completes successfully. Also check on the EC2 instance host to see the log volume mounted and also the mount point path.

6. Copy the archive logs from the mounted log volume to the current archive log directory.

   [ec2-user@ip-10-0-0-151 ~]$ cp /var/opt/snapcenter/sco/backup_mount/ip-10-0-0-151_03-25-2022_11.15.01.1503_1/ORCL/1/db/ORCL_A/arch/*.arc /ora_nfs_log/db/ORCL_A/arch/

7. Return to the SnapCenter Resource tab > database backup page, highlight the data snapshot copy, and click the Restore button to start the database restore workflow.
8. Check "All Datafiles" and "Change database state if needed for restore and recovery", and click Next.

9. Choose a desired recovery scope using either SCN or time. Rather than copying the mounted archive logs
to the current log directory as demonstrated in step 6, the mounted archive log path can be listed in "Specify external archive log files locations" for recovery.

10. Specify an optional prescript to run if necessary.
11. Specify an optional afterscript to run if necessary. Check the open database after recovery.
12. Provide an SMTP server and email address if a job notification is needed.
13. Restore the job summary. Click finish to launch the restore job.
14. Validate the restore from SnapCenter.
15. Validate the restore from the EC2 instance host.
16. To unmount the restore log volume, reverse the steps in step 4.

Creating a database clone

The following section demonstrates how to use the SnapCenter clone workflow to create a database clone from a primary database to a standby EC2 instance.

1. Take a full snapshot backup of the primary database from SnapCenter using the full backup resource group.

2. From the SnapCenter Resource tab > Database view, open the Database Backup Management page for the primary database that the replica is to be created from.

3. Mount the log volume snapshot taken in step 4 to the standby EC2 instance host.
4. Highlight the snapshot copy to be cloned for the replica, and click the Clone button to start the clone procedure.
5. Change the replica copy name so that it is different from the primary database name. Click Next.

6. Change the clone host to the standby EC2 host, accept the default naming, and click Next.
7. Change your Oracle home settings to match those configured for the target Oracle server host, and click Next.
8. Specify a recovery point using either time or the SCN and mounted archive log path.
9. Send the SMTP email settings if needed.
10. Clone the job summary, and click Finish to launch the clone job.
11. Validate the replica clone by reviewing the clone job log.
The cloned database is registered in SnapCenter immediately.
12. Turn off Oracle archive log mode. Log into the EC2 instance as oracle user and execute following command:

```sql
sqlplus / as sysdba
shutdown immediate;
startup mount;
alter database noarchivelog;
alter database open;
```

Instead primary Oracle backup copies, a clone can also be created from replicated secondary backup copies on target FSx cluster with same procedures.

**HA failover to standby and resync**

The standby Oracle HA cluster provides high availability in the event of failure in the primary site, either in the compute layer or in the storage layer. One significant benefit of the solution is that a user can test and validate the infrastructure at any time or with any frequency. Failover can be user simulated or triggered by real failure. The failover processes are identical and can be automated for fast application recovery.

See the following list of failover procedures:

1. For a simulated failover, run a log snapshot backup to flush the latest transactions to the standby site, as demonstrated in the section [Taking an archive log snapshot](#). For a failover triggered by an actual failure, the last recoverable data is replicated to the standby site with the last successful scheduled log volume backup.

2. Break the SnapMirror between primary and standby FSx cluster.

3. Mount the replicated standby database volumes at the standby EC2 instance host.

4. Relink the Oracle binary if the replicated Oracle binary is used for Oracle recovery.

5. Recover the standby Oracle database to the last available archive log.

6. Open the standby Oracle database for application and user access.

7. For an actual primary site failure, the standby Oracle database now takes the role of the new primary site and database volumes can be used to rebuild the failed primary site as a new standby site with the reverse SnapMirror method.

8. For a simulated primary site failure for testing or validation, shut down the standby Oracle database after the completion of testing exercises. Then unmount the standby database volumes from the standby EC2 instance host and resync replication from the primary site to the standby site.
These procedures can be performed with the NetApp Automation Toolkit available for download at the public NetApp GitHub site.

```
git clone https://github.com/NetApp-Automation/na_ora_hadr_failover_resync.git
```

Read the README instruction carefully before attempting setup and failover testing.

**Database migration from on-prem to public cloud**

Database migration is a challenging endeavor by any means. Migrating an Oracle database from on-premises to the cloud is no exception.

The following sections provide key factors to consider when migrating Oracle databases to the AWS public cloud with the AWS EC2 compute and FSx storage platform.

**ONTAP storage is available on-premises**

If the on-premises Oracle database is sitting on an ONTAP storage array, then it is easier to set up replication for database migration using the NetApp SnapMirror technology that is built into AWS FSx ONTAP storage. The migration process can be orchestrated using NetApp BlueXP console.

1. Build a target compute EC2 instance that matches the on-premises instance.
2. Provision matching, equally sized database volumes from FSx console.
3. Mount the FSx database volumes to the EC2 instance.
4. Set up SnapMirror replication between the on-premises database volumes to the target FSx database volumes. The initial sync might take some time to move the primary source data, but any following incremental updates are much quicker.
5. At the time of switchover, shut down the primary application to stop all transactions. From the Oracle sqlplus CLI interface, execute an Oracle online log switch and allow SnapMirror sync to push the last archived log to the target volume.
6. Break up the mirrored volumes, run Oracle recovery at the target, and bring up the database for service.
7. Point applications to the Oracle database in the cloud.

The following video demonstrates how to migrate an Oracle database from on-premises to AWS FSx/EC2 using the NetApp BlueXP console and SnapMirror replication.

[Video: Migrate on-prem Oracle DB to AWS]

**ONTAP storage is not available on-premises**

If the on-premises Oracle database is hosted on third-party storage other than ONTAP, database migration is based on the restore of a Oracle database backup copy. You must play the archive log to make it current before switching over.

AWS S3 can be used as a staging storage area for database move and migration. See the following high level steps for this method:

1. Provision a new, matching EC2 instance that is comparable with the on-premises instance.
2. Provision equal database volumes from FSx storage and mount the volumes to the EC2 instance.
3. Create a disk-level Oracle backup copy.
4. Move the backup copy to AWS S3 storage.
5. Recreate the Oracle control file and restore and recover the database by pulling data and the archive log from S3 storage.
6. Sync the target Oracle database with the on-premises source database.
7. At switchover, shut down the application and source Oracle database. Copy the last few archive logs and apply them to the target Oracle database to bring it up to date.
8. Start up the target database for user access.
9. Redirect application to the target database to complete the switchover.

**Migrate on-premises Oracle databases to AWS FSx/EC2 using PDB relocation with maximum availability**

This migration approach is best suited to Oracle databases that are already deployed in PDB/CDB multitenant model, and ONTAP storage is not available on-premises. The PDB relocation method utilizes Oracle PDB hot clone technology to move PDBs between a source CDB and a target CDB while minimizing service interruption.

First, create CDB in the AWS FSx/EC2 with sufficient storage to host PDBs to be migrated from on-premises. Multiple on-premises PDBs can be relocated one at a time.

1. If the on-premises database is deployed in a single instance rather than in the multitenant PDB/CDB model, follow the instructions in [Converting a single instance non-CDB to a PDB in a multitenant CDB](#) to convert the single instance to multitenant PDB/CDB. Then follow the next step to migrate the converted PDB to CDB in AWS FSx/EC2.
2. If the on-premises database is already deployed in the multitenant PDB/CDB model, follow the instructions in [Migrate on-premises Oracle databases to cloud with PDB relocation](#) to perform the migration.

The following video demonstrates how an Oracle database (PDB) can be migrated to FSx/EC2 using PDB relocation with maximum availability.

**Migrate on-prem Oracle PDB to AWS CDB with max availability**

> Although the instructions in step 1 and 2 are illustrated in the context of Azure public cloud, the procedures are applicable to AWS cloud without any changes.

The NetApp Solutions Automation team provides a migration toolkit that can facilitate Oracle database migration from on-premises to the AWS cloud. Use following command to download the Oracle database migration toolkit for PDB relocation.

```
git clone https://github.com/NetApp-Automation/na_ora_aws_migration.git
```

**Oracle Database Deployment and Protection on Azure NetApp Files**

**TR-4954: Oracle Database Deployment and Protection on Azure NetApp Files**

Author(s): Allen Cao, Niyaz Mohamed, NetApp
Overview

Many mission-critical Oracle enterprise databases are still hosted on-premises, and many enterprises are looking to migrate these Oracle databases to a public cloud. Often, these Oracle databases are application centric and thus require user-specific configurations, a capability that is missing from many database-as-a-service public-cloud offerings. Therefore, the current database landscape calls for a public-cloud-based Oracle database solution built from a high-performance, scalable compute and storage service that can accommodate unique requirements. Azure virtual machine compute instances and the Azure NetApp Files storage service might be the missing pieces of this puzzle that you can leverage to build and migrate your mission-critical Oracle database workloads to a public cloud.

Azure Virtual Machine

Azure virtual machines are one of several types of on-demand, scalable computing resources that Azure offers. Typically, you choose a virtual machine when you need more control over the computing environment than the other choices offer. Azure virtual machines offer a quick and easy way to create a computer with specific configurations required to run your Oracle database, whether it is for compute- or memory-intensive workloads. Virtual machines in an Azure virtual network can easily be connected to your organization’s network, for example through a secured VPN tunnel.

Azure NetApp Files (ANF)

Azure NetApp Files is a fully managed Microsoft service that will take your database workload to the cloud faster and more securely than ever before. It was designed to meet the core requirements of running high-performance workloads such as Oracle databases in the cloud, and it provides performance tiers that reflect the real-world range of IOPS demands, low latency, high availability, high durability, manageability at scale, and fast and efficient backup, recovery, and cloning. These capabilities are possible because Azure NetApp Files is based on physical all-flash NetApp ONTAP systems running within the Azure data center environment. Azure NetApp Files is completely integrated into the Azure DCs and portal, and customers can use the same comfortable graphical interface and APIs for creating and managing shared files as with any other Azure object. With Azure NetApp file, you can unlock the full capabilities of Azure without extra risk, cost, or time and trust the only enterprise file service native to Azure.

Conclusion

This documentation describes in detail how to deploy, configure, and protect an Oracle database with an Azure virtual machine and Azure NetApp Files storage service that delivers performance and durability similar to an on-premises system. For best-practices guidance, see TR-4780 Oracle Databases on Microsoft Azure. More importantly, NetApp also provides automation toolkits that automate most of the tasks that are required for the deployment, configuration, data protection, migration, and management of your Oracle database workload in the Azure public cloud. The automation toolkits are available for download at NetApp public GitHub site: NetApp-Automation.

Solution Architecture

The following architecture diagram illustrates a highly available Oracle database deployment on Azure VM instances and the Azure NetApp Files storage.

Within the environment, the Oracle compute instance is deployed via an Azure services VM console. There are multiple Azure instance types available from the console. NetApp recommends deploying a database-oriented Azure VM instance that meets your expected workload.

Oracle database storage on the other hand is deployed with the Azure NetApp Files service available from Azure console. The Oracle binary, data, or log volumes are subsequently presented and mounted on an Azure VM instance Linux host.
In many respects, the implementation of Azure NetApp Files in Azure cloud is very similar to an on-premises ONTAP data storage architecture with many built-in redundancies, such as RAID and dual controllers. For disaster recovery, a standby site can be setup in different regions and database can be synced up with the primary site using application-level replication (for example, Oracle Data Guard).

In our test validation for Oracle database deployment and data protection, the Oracle database is deployed on a single Azure VM as illustrated in the following diagram:
The Azure Oracle environment can be managed with an Ansible controller node for automation using tool kits provided by NetApp for database deployment, backup, recovery, and database migration. Any updates to the
Oracle Azure VM instance operating-system kernel or Oracle patching can be performed in parallel to keep the primary and standby in sync. In fact, the initial toolkits can be easily expanded to perform daily Oracle tasks if needed. If you need help to set up a CLI Ansible controller, see NetApp Solution Automation to get started.

**Factors to consider for Oracle database deployment**

A public cloud provides many choices for compute and storage, and using the correct type of compute instance and storage engine is a good place to start for database deployment. You should also select compute and storage configurations that are optimized for Oracle databases.

The following sections describe the key considerations when deploying an Oracle database in the Azure public cloud on an Azure virtual machine instance with Azure NetApp Files storage.

**VM type and sizing**

Selecting the right VM type and size is important for optimal performance of a relational database in a public cloud. An Azure virtual machine provides a variety of compute instances that can be used to host Oracle database workloads. See the Microsoft documentation [Sizes for virtual machines in Azure](https://docs.microsoft.com/en-us/azure/virtual-machines/sizes) for different types of Azure virtual machines and their sizing. In general, NetApp recommends using a general-purpose Azure virtual machine for the deployment of small- and medium-sized Oracle databases. For the deployment of larger Oracle databases, a memory-optimized Azure VM is appropriate. With more available RAM, a larger Oracle SGA or smart flash cache can be configured to reduce the physical I/O, which in turn improves database performance.

Azure NetApp Files works as an NFS mount attached to an Azure virtual machine, which offers higher throughput and overcomes the storage-optimized VM throughput limit with local storage. Therefore, running Oracle on Azure NetApp Files could reduce the licensable Oracle CPU core count and licensing costs. See [TR-4780: Oracle Databases on Microsoft Azure](https://docs.microsoft.com/en-us/trustcenter/tr-4780), Section 7 - How Does Oracle Licensing Work?

Other factors to consider include the following:

- Choose the correct vCPU and RAM combination based on workload characteristics. As the RAM size increases on the VM, so does the number of vCPU cores. There should be a balance at some point as the Oracle license fees are charged on the number of vCPU cores.
- Add swap space to a VM. The default Azure VM deployment does not create a swap space, which is not optimal for a database.

**Azure NetApp Files performance**

Azure NetApp Files volumes are allocated from a capacity pool the customer must provision in their Azure NetApp Files storage account. Each capacity pool is assigned as follows:

- To a service level that defines the overall performance capability.
- The initially provisioned storage capacity or tiering for that capacity pool. A quality of service (QoS) level that defines the overall maximum throughput per provisioned space.

The service level and initially provisioned storage capacity determines the performance level for a particular Oracle database volume.

1. **Service Levels for Azure NetApp Files**

• **Ultra storage.** This tier provides up to 128MiBps of throughput per 1TiB of volume quota assigned.
• **Premium storage.** This tier provides up to 64MiBps of throughput per 1TiB of volume quota assigned.
• **Standard storage.** This tier provides up to 16MiBps of throughput per 1TiB of volume quota assigned.

### 2. Capacity pool and quality of service

Each of the desired service levels has an associated cost for provisioned capacity and includes a quality-of-service (QoS) level that defines the overall maximum throughput for provisioned space.

For example, a 10TiB-provisioned single-capacity pool with the premium service level provides an overall available throughput for all volumes in this capacity pool of 10x 64MBps, so 640MBps with 40,000 (16K) IOPs or 80,000 (8K) IOPs.

The minimum capacity pool size is 4TiB. You can change the size of a capacity pool in 1TiB increments in response to changes in your workload requirements to manage storage needs and costs.

### 3. Calculate the service level at a database volume

The throughput limit for an Oracle database volume is determined by a combination of the following factors:
The service level of the capacity pool to which the volume belongs and The quota assigned to the volume.

The following diagram shows how the throughput limit for an Oracle database volume is calculated.

In example 1, a volume from a capacity pool with the Premium storage tier that is assigned 2TiB of quota is assigned a throughput limit of 128MiBps (2TiB * 64MiBps). This scenario applies regardless of the capacity pool size or the actual volume consumption.

In example 2, a volume from a capacity pool with the Premium storage tier that is assigned 100GiB of quota is assigned a throughput limit of 6.25MiBps (0.09765625TiB * 64MiBps). This scenario applies regardless of the capacity pool size or the actual volume consumption.

Please note that the minimum volume size is 100GiB.

### Storage layout and settings

NetApp recommends the following storage layout:
• For small databases, using single volume layout for all Oracle files.

• For large databases, the recommended volume layout is multiple volumes: one for Oracle data and a duplicate control file and one for the Oracle active log, archived log, and control file. NetApp highly recommends allocating a volume for the Oracle binary instead of the local drive so that the database can be relocated to a new host and quickly restored.
NFS configuration

Linux, the most common operating system, includes native NFS capabilities. Oracle offers a direct NFS (dNFS) client natively integrated into Oracle. Oracle dNFS bypasses the OS cache and enables parallel processing to improve database performance. Oracle has supported NFSv3 for over 20 years, and NFSv4 is supported with Oracle 12.1.0.2 and later.

By using dNFS (available since Oracle 11g), an Oracle database running on an Azure Virtual Machine can drive significantly more I/O than the native NFS client. Automated Oracle deployment using the NetApp automation toolkit automatically configures dNFS on NFSv3.

The following diagram demonstrates the SLOB benchmark on Azure NetApp Files with Oracle dNFS.

Other factors to consider:

- TCP slot tables are the NFS equivalent of host-bus-adapter (HBA) queue depth. These tables control the number of NFS operations that can be outstanding at any one time. The default value is usually 16, which is far too low for optimum performance. The opposite problem occurs on newer Linux kernels, which can automatically increase the TCP slot table limit to a level that saturates the NFS server with requests.

  For optimum performance and to prevent performance problems, adjust the kernel parameters that control TCP slot tables to 128.

```
  sysctl -a | grep tcp.*.slot_table
```

- The following table provides recommended NFS mount options for a single instance of Linux NFSv3.
Before using dNFS, verify that the patches described in Oracle Doc 1495104.1 are installed. The NetApp Support matrix for NFSv3 and NFSv4 do not include specific operating systems. All OSs that obey the RFC are supported. When searching the online IMT for NFSv3 or NFSv4 support, do not select a specific OS because no matches will be displayed. All OSs are implicitly supported by the general policy.

### Step-by-Step Oracle deployment procedures on Azure VM and Azure NetApp Files

**Deploy an Azure VM with ANF for Oracle via Azure portal console**

If you are new to Azure, you first need to set up an Azure account environment. This includes signing up your organization to use Azure Active Directory. The following section is a summary of these steps. For details, see the linked Azure-specific documentation.

**Create and consume Azure resources**

After your Azure environment is set up and an account is created and associated with a subscription, you can log into Azure portal with the account to create the necessary resources to run Oracle.

1. **Create a virtual network or VNet**

   Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VMs), to securely communicate with each other, the internet, and on-premises networks. Before provisioning an Azure VM, a VNet (where a VM is deployed) must first be configured.

   See [Create a virtual network using the Azure portal](#) to create a VNet.

2. **Create a NetApp storage account and capacity pool for ANF**

   In this deployment scenario, an Azure VM OS is provisioned using regular Azure storage, but ANF volumes are provisioned to run Oracle database via NFS. First, you need to create a NetApp storage account and a capacity pool to host the storage volumes.

   See [Set up Azure NetApp Files and create an NFS volume](#) to set up an ANF capacity pool.

3. **Provision Azure VM for Oracle**

   Based on your workload, determine what type of Azure VM you need and the size of the VM vCPU and RAM to deploy for Oracle. Then, from the Azure console, click the VM icon to launch the VM deployment workflow.

   1. From the Azure VM page, click [Create] and then choose [Azure virtual machine].

<table>
<thead>
<tr>
<th>File Type</th>
<th>Mount Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control files</td>
<td>rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536</td>
</tr>
<tr>
<td>Data files</td>
<td></td>
</tr>
<tr>
<td>Redo logs</td>
<td></td>
</tr>
<tr>
<td>ORACLE_HOME</td>
<td>rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536</td>
</tr>
<tr>
<td>ORACLE_BASE</td>
<td></td>
</tr>
</tbody>
</table>
2. Choose the subscription ID for the deployment, and then choose the resource group, region, host name, VM image, size, and authentication method. Go to the Disk page.
Create a virtual machine

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. Learn more

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription
Hybrid Cloud TME Onprem

Resource group
ANFAVSRG

Instance details

Virtual machine name
aco-ora01

Region
(US) South Central US

Availability options
No infrastructure redundancy required

Security type
Standard

Image
Red Hat Enterprise Linux 8.0 (LVM) Gen2

Run with Azure Spot discount

Size
Standard_D8s_v3 - 8 vcpus, 32 GiB memory ($273.02/month)

Administrator account

Authentication type
SSH public key

Password

Review + create
3. Choose premium SSD for OS local redundancy and leave the data disk blank because the data disks are mounted from ANF storage. Go to the Networking page.
4. Choose the VNet and subnet. Allocate a public IP for external VM access. Then go to the Management page.
5. Keep all defaults for Management and move to the Advanced page.
6. Keep all defaults for the Advanced page unless you need to customize a VM after deployment with custom scripts. Then go to Tags page.
7. Add a tag for the VM if desired. Then, go to the Review + create page.
8. The deployment workflow runs a validation on the configuration, and, if the validation passes, click **Create** to create the VM.

4. **Provision ANF database volumes for Oracle**

You must create three NFS volumes for an ANF capacity pool for the Oracle binary, data, and log volumes respectively.
1. From the Azure console, under the list of Azure services, click Azure NetApp Files to open a volume creation workflow. If you have more than one ANF storage account, click the account that you would like to provision volumes from.

2. Under your NetApp storage account, click **Volumes**, and then **Add volume** to create new Oracle volumes.
3. As a good practice, identify Oracle volumes with the VM hostname as a prefix and then followed by the mount point on the host, such as u01 for Oracle binary, u02 for Oracle data, and u03 for Oracle log. Choose the same VNet for the volume as for the VM. Click **Next: Protocol**.

4. Choose the NFS protocol, add the Oracle host IP address to the allowed client, and remove the default policy that allows all IP addresses 0.0.0.0/0. Then click **Next: Tags**.
5. Add a volume tag if desired. Then click **Review + Create**.
6. If the validation passes, click **Create** to create the volume.
Install and configure Oracle on Azure VM with ANF

The NetApp solutions team has created many Ansible-based automation toolkits to help you deploy Oracle in Azure smoothly. Follow these steps to deploy Oracle on an Azure VM.

Set up an Ansible controller

If you have not set up an Ansible controller, see NetApp Solution Automation, which has detailed instructions on how to setup an Ansible controller.

Obtain Oracle deployment automation toolkit

Clone a copy of the Oracle deployment toolkit in your home directory under the user ID that you use to log into the Ansible controller.

```
git clone https://github.com/NetApp-Automation/na_oracle19c_deploy.git
```

Execute the toolkit with your configuration

See the CLI deployment Oracle 19c Database to execute the playbook with the CLI. You can ignore the ONTAP portion of the variables configuration in the global VARS file when you create database volumes from
the Azure console rather than the CLI.

The toolkit default deploys Oracle 19c with RU 19.8. It can be easily adapted for any other patch level with minor default configuration changes. Also default seed-database active log files are deployed into the data volume. If you need active log files on the log volume, it should be relocated after initial deployment. Reach out to the NetApp Solution team for help if needed.

Set up AzAcSnap backup tool for app-consistent snapshots for Oracle

The Azure Application-Consistent Snapshot tool (AzAcSnap) is a command-line tool that enables data protection for third-party databases by handling all the orchestration required to put them into an application-consistent state before taking a storage snapshot. It then returns these databases to an operational state. NetApp recommends installing the tool on the database server host. See the following installation and configuration procedures.

Install AzAcSnap tool

1. Get the most recent version of the AzArcSnap Installer.
2. Copy the downloaded self-installer to the target system.
3. Execute the self-installer as the root user with the default installation option. If necessary, make the file executable using the `chmod +x *.run` command.

   ```bash
   ./azacsnap_installer_v5.0.run -I
   ```

Configure Oracle connectivity

The snapshot tools communicate with the Oracle database and need a database user with appropriate permissions to enable or disable backup mode.

1. Set up AzAcSnap database user

   The following examples show the setup of the Oracle database user and the use of sqlplus for communication to the Oracle database. The example commands set up a user (AZACSNAP) in the Oracle database and change the IP address, usernames, and passwords as appropriate.

   1. From the Oracle database installation, launch sqlplus to log into the database.

      ```bash
      su - oracle
      sqlplus / AS SYSDBA
      ```

   2. Create the user.

      ```sql
      CREATE USER azacsnap IDENTIFIED BY password;
      ```

   3. Grant the user permissions. This example sets the permission for the AZACSNAP user to enable putting the database into backup mode.
GRANT CREATE SESSION TO azacsnap;
GRANT SYSBACKUP TO azacsnap;

4. Change the default user’s password expiration to unlimited.

ALTER PROFILE default LIMIT PASSWORD_LIFE_TIME unlimited;

5. Validate azacsnap connectivity for the database.

connect azacsnap/password
quit;

2. Configure Linux-user azacsnap for DB access with Oracle wallet

The AzAcSnap default installation creates an azacsnap OS user. Its Bash shell environment must be
configured for Oracle database access with the password stored in an Oracle wallet.

1. As root user, run the cat /etc/oratab command to identify the ORACLE_HOME and ORACLE_SID
variables on the host.

cat /etc/oratab

2. Add ORACLE_HOME, ORACLE_SID, TNS_ADMIN, and PATH variables to the azacsnap user bash
profile. Change the variables as needed.

```bash
echo "export ORACLE_SID=ORATEST" >> /home/azacsnap/.bash_profile
echo "export ORACLE_HOME=/u01/app/oracle/product/19800/ORATST" >>
/home/azacsnap/.bash_profile
echo "export TNS_ADMIN=/home/azacsnap" >> /home/azacsnap/.bash_profile
echo "export PATH=$PATH:$ORACLE_HOME/bin" >>
/home/azacsnap/.bash_profile
```

3. As the Linux user azacsnap, create the wallet. You are prompted for the wallet password.

```bash
sudo su - azacsnap
mkstore -wrl $TNS_ADMIN/.oracle_wallet/ -create
```

4. Add the connect string credentials to the Oracle Wallet. In the following example command, AZACSNAP is
the ConnectString to be used by AzAcSnap, azacsnap is the Oracle Database User, and AzPasswd1 is the
Oracle User’s database password. You are again prompted for the wallet password.
Configure ANF connectivity

This section explains how to enable communication with Azure NetApp Files (with a VM).
1. Within an Azure Cloud Shell session, make sure that you are logged into the subscription that you want to be associated with the service principal by default.

   `az account show`

2. If the subscription isn't correct, use the following command:

   `az account set -s <subscription name or id>`

3. Create a service principal using the Azure CLI as in the following example:

   `az ad sp create-for-rbac --name "AzAcSnap" --role Contributor --scopes /subscriptions/{subscription-id} --sdk-auth`

   The expected output:

   ```
   {
     "clientId": "00aa000a-aaaa-0000-00a0-00aa000aaa0a",
     "clientSecret": "00aa000a-aaaa-0000-00a0-00aa000aaa0a",
     "subscriptionId": "00aa000a-aaaa-0000-00a0-00aa000aaa0a",
     "tenantId": "00aa000a-aaaa-0000-00a0-00aa000aaa0a",
     "activeDirectoryEndpointUrl": "https://login.microsoftonline.com",
     "resourceManagerEndpointUrl": "https://management.azure.com/",
     "activeDirectoryGraphResourceId": "https://graph.windows.net/",
     "sqlManagementEndpointUrl": "https://management.core.windows.net:8443/",
     "galleryEndpointUrl": "https://gallery.azure.com/",
     "managementEndpointUrl": "https://management.core.windows.net/"
   }
   ```

4. Cut and paste the output content into a file called `oracle.json` stored in the Linux user azacsnap user bin directory and secure the file with the appropriate system permissions.

   Make sure the format of the JSON file is exactly as described above, especially with the URLs enclosed in double quotes (").

Complete the setup of AzAcSnap tool

Follow these steps to configure and test the snapshot tools. After successful testing, you can perform the first database-consistent storage snapshot.

1. Change into the snapshot user account.
su - azacsnap

2. Change the location of commands.

   cd /home/azacsnap/bin/

3. Configure a storage backup detail file. This creates an azacsnap.json configuration file.

   azacsnap -c configure --configuration new

The expected output with three Oracle volumes:

   [azacsnap@acao-ora01 bin]$ azacsnap -c configure --configuration new
   Building new config file
   Add comment to config file (blank entry to exit adding comments): Oracle snapshot bkup
   Add comment to config file (blank entry to exit adding comments):
   Enter the database type to add, 'hana', 'oracle', or 'exit' (for no database): oracle

   === Add Oracle Database details ===
   Oracle Database SID (e.g. CDB1): ORATST
   Database Server's Address (hostname or IP address): 172.30.137.142
   Oracle connect string (e.g. /@AZACSNAP): /@AZACSNAP

   === Azure NetApp Files Storage details ===
   Are you using Azure NetApp Files for the database? (y/n) [n]: y
   --- DATA Volumes have the Application put into a consistent state before they are snapshot ---
   Add Azure NetApp Files resource to DATA Volume section of Database configuration? (y/n) [n]: y
   Full Azure NetApp Files Storage Volume Resource ID (e.g. /subscriptions/.../resourceGroups/.../providers/Microsoft.NetApp/netAppAccounts/.../capacityPools/Premium/volumes/...): /subscriptions/0efa2dfb-917c-4497-b56a-b3f4eadb8111/resourceGroups/ANFAVSRG/providers/Microsoft.NetApp/netAppAccounts/ANFAVSAcct/capacityPools/CapPool/volumes/acao-ora01-u01
   Service Principal Authentication filename or Azure Key Vault Resource ID (e.g. auth-file.json or https://.../): oracle.json
   Add Azure NetApp Files resource to DATA Volume section of Database configuration? (y/n) [n]: y
   Full Azure NetApp Files Storage Volume Resource ID (e.g. /subscriptions/.../resourceGroups/.../providers/Microsoft.NetApp/netAppAccounts/.../capacityPools/Premium/volumes/...): /subscriptions/0efa2dfb-917c-4497-b56a-b3f4eadb8111/resourceGroups/ANFAVSRG/providers/Microsoft.NetApp/netAppAccounts/ANFAVSAcct/capacityPools/CapPool/volumes/acao-ora01-u01
   Service Principal Authentication filename or Azure Key Vault Resource ID (e.g. auth-file.json or https://.../): oracle.json
   Add Azure NetApp Files resource to DATA Volume section of Database configuration? (y/n) [n]: y
4. As the azacsnap Linux user, run the azacsnap test command for an Oracle backup.

```bash
cd ~/bin
azacsnap -c test --test oracle --configfile azacsnap.json
```

The expected output:
5. Run your first snapshot backup.

```bash
azacsnap -c backup --volume data --prefix ora_test --retention=1
```

Protect your Oracle database in Azure cloud

Allen Cao, NetApp Solutions Engineering

This section describes how to protect your Oracle database with azacsnap tool and snapshot backup, restore and snapshots tiering to Azure blob.

**Backup Oracle database with snapshot using AzAcSnap tool**

The Azure Application-Consistent Snapshot tool (AzAcSnap) is a command-line tool that enables data protection for third-party databases by handling all the orchestration required to put them into an application-consistent state before taking a storage snapshot, after which it returns the databases to an operational state.

In the case of Oracle, you put the database in backup mode to take a snapshot and then take the database out of backup mode.

**Backup data and log volumes**

The backup can be set up on the database server host with simple shell script that executes the snapshot command. Then, the script can be scheduled to run from crontab.

Generally, the frequency of backup depends on the desired RTO and RPO. Frequent snapshot creation consumes more storage space. There is a trade off between the frequency of backup and space consumption.

Data volumes typically consume more storage space than log volumes. Therefore, you can take snapshots on data volumes every few hours and more frequent snapshots on log volumes every 15 to 30 minutes.

See the following examples of backup scripts and scheduling.

For data volume snapshots:
# /bin/sh
cd /home/azacsnap/bin
. ~/.bash_profile
azacsnap -c backup --volume data --prefix acao-ora01-data --retention 36
azacsnap -c backup --volume other --prefix acao-ora01-log --retention 250

For log volume snapshots:

# /bin/sh
cd /home/azacsnap/bin
. ~/.bash_profile
azacsnap -c backup --volume other --prefix acao-ora01-log --retention 250

Crontab schedule:

15,30,45 * * * * /home/azacsnap/snap_log.sh
0 */2 * * * /home/azacsnap/snap_data.sh

When setting up the backup azacsnap.json configuration file, add all data volumes, including
the binary volume, to dataVolume and all log volumes to otherVolume. The maximum
retention of snapshots is 250 copies.

Validate the snapshots

Go to the Azure portal > Azure NetApp Files/volumes to check if the snapshots have been successfully
created.
Oracle restore and recovery from local backup

One of key benefits of snapshot backup is that it coexists with source database volumes, and the primary database volumes can be rolled back almost instantly.

Restore and recovery of Oracle on the primary server

The following example demonstrates how to restore and recover an Oracle database from the Azure dashboard and CLI on the same Oracle host.

1. Create a test table in the database to be restored.
[oracle@acoo-ora01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Mon Sep 12 19:02:35 2022
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL> create table testsnapshot(
   2     id integer,
   3     event varchar(100),
   4     dt timestamp);

Table created.

SQL> insert into testsnapshot values(1,'insert a data marker to validate
snapshot restore',sysdate);

1 row created.

SQL> commit;

Commit complete.

SQL> select * from testsnapshot;

    ID  
---

    EVENT
-----------------------------------------------

-----------------------------------------------

    DT
-----------------------------------------------

----------

1

insert a data marker to validate snapshot restore
12-SEP-22 07.07.35.000000 PM

2. Drop the table after the snapshot backups.
[oracle@acao-ora01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Tue Sep 13 14:20:22 2022
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL> drop table testsnapshot;

Table dropped.

SQL> select * from testsnapshot;
select * from testsnapshot
  *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

3. From the Azure NetApp Files dashboard, restore the log volume to the last available snapshot. Choose Revert volume.
4. Confirm revert volume and click Revert to complete the volume reversion to the latest available backup.

5. Repeat the same steps for the data volume, and make sure that the backup contains the table to be recovered.

6. Again confirm the volume reversion, and click "Revert."
7. Resync the control files if you have multiple copies of them, and replace the old control file with the latest copy available.

```
[oracle@acao-ora01 ~]$ mv /u02/oradata/ORATST/control01.ctl /u02/oradata/ORATST/control01.ctl.bk
[oracle@acao-ora01 ~]$ cp /u03/orareco/ORATST/control02.ctl /u02/oradata/ORATST/control01.ctl
```

8. Log into the Oracle server VM and run database recovery with sqlplus.

```
[oracle@acao-ora01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Tue Sep 13 15:10:17 2022
Version 19.8.0.0.0
Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup mount;
ORACLE instance started.

Total System Global Area 6442448984 bytes
Fixed Size 8910936 bytes
Variable Size 1090519040 bytes
Database Buffers 5335154688 bytes
Redo Buffers 7864320 bytes
Database mounted.
```
SQL> recover database using backup controlfile until cancel;
ORA-00279: change 3188523 generated at 09/13/2022 10:00:09 needed for thread 1
ORA-00289: suggestion :
/u03/orareco/ORATST/archivelog/2022_09_13/o1_mf_1_43__22rnjq9q_.arc
ORA-00280: change 3188523 for thread 1 is in sequence #43

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 3188862 generated at 09/13/2022 10:01:20 needed for thread 1
ORA-00289: suggestion :
/u03/orareco/ORATST/archivelog/2022_09_13/o1_mf_1_44__29f2lgb5_.arc
ORA-00280: change 3188862 for thread 1 is in sequence #44
ORA-00278: log file
'/u03/orareco/ORATST/archivelog/2022_09_13/o1_mf_1_43__22rnjq9q_.arc' no longer needed for this recovery

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 3193117 generated at 09/13/2022 12:00:08 needed for thread 1
ORA-00289: suggestion :
/u03/orareco/ORATST/archivelog/2022_09_13/o1_mf_1_45__29h6qwyw_.arc
ORA-00280: change 3193117 for thread 1 is in sequence #45
ORA-00278: log file
'/u03/orareco/ORATST/archivelog/2022_09_13/o1_mf_1_43__22rnjq9q_.arc' no longer needed for this recovery

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 3193440 generated at 09/13/2022 12:01:20 needed for thread 1
ORA-00289: suggestion :
/u03/orareco/ORATST/archivelog/2022_09_13/o1_mf_1_46_%u_.arc
ORA-00280: change 3193440 for thread 1 is in sequence #46
ORA-00278: log file
'/u03/orareco/ORATST/archivelog/2022_09_13/o1_mf_1_45__29h6qwyw_.arc' no longer needed for this recovery

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
cancel
Media recovery cancelled.
This screen demonstrates that the dropped table has been recovered using local snapshot backups.

**Database migration from on-premises to Azure cloud**

As a result of the Oracle decision to phase out single-instance databases, many organizations have converted single-instance Oracle databases to multitenant container databases. This enables the easy relocation of a subset of container databases called PDB to cloud with the maximum availability option, which minimize downtime during migration.

However, if you still have a single instance of a Oracle database, it can first be converted into a multitenant container database in place before attempting PDB relocation.

The following sections provide details for the migration of on-premises Oracle databases to Azure cloud in either scenarios.

**Converting a single instance non-CDB to a PDB in a multitenant CDB**

If you still have a single-instance Oracle database, it must be converted into a multitenant container database whether you wish to migrate it to the cloud or not, because Oracle will stop supporting single-instance databases some time soon.

The following procedures plug a single instance database into a container database as a pluggable database
1. Build a shell container database on the same host as the single-instance database in a separate ORACLE_HOME.

2. Shut down the single instance database and restart it in read-only mode.

3. Run the DBMS_PDB.DESCRIBE procedure to generate the database metadata.

```sql
BEGIN
    DBMS_PDB.DESCRIBE(
        pdb_descr_file => '/home/oracle/ncdb.xml');
END;
/
```

4. Shut down the single-instance database.

5. Start up the container database.

6. Run the DBMS_PDB.CHECK_PLUG_COMPATIBILITY function to determine whether the non-CDB is compatible with the CDB.

```sql
SET SERVEROUTPUT ON
DECLARE
    compatible CONSTANT VARCHAR2(3) :=
        CASE DBMS_PDB.CHECK_PLUG_COMPATIBILITY(
            pdb_descr_file => '/disk1/oracle/ncdb.xml',
            pdb_name       => 'NCDB')
            WHEN TRUE THEN 'YES'
            ELSE 'NO'
        END;
BEGIN
    DBMS_OUTPUT.PUT_LINE(compatible);
END;
/
```

If the output is YES, then the non-CDB is compatible, and you can continue with the next step.

If the output is NO, then the non-CDB is not compatible, and you can check the PDB_PLUG_IN_VIOLATIONS view to see why it is not compatible. All violations must be corrected before you continue. For example, any version or patch mismatches should be resolved by running an upgrade or the opatch utility. After correcting the violations, run DBMS_PDB.CHECK_PLUG_COMPATIBILITY again to ensure that the non-CDB is compatible with the CDB.

7. Plug in the single instance non-CDB.
CREATE PLUGGABLE DATABASE ncdb USING '/home/oracle/ncdb.xml'
COPY
    FILE_NAME_CONVERT = ('/disk1/oracle/dbs/', '/disk2/oracle/ncdb/')
;

If there is not sufficient space on the host, the NOCOPY option can be used to create the PDB. In that case, a single-instance non-CDB is not useable after plug in as a PDB because the original data files has been used for the PDB. Make sure to create a backup before the conversion so that there is something to fall back on if anything goes wrong.

8. Start with PDB upgrade after conversion if the version between the source single-instance non-CDB and the target CDB are different. For the same-version conversion, this step can be skipped.

```sql
sqlplus / as sysdba;
alter session set container=ncdb
alter pluggable database open upgrade;
exit;
dbupgrade -c ncdb -l /home/oracle
```

Review the upgrade log file in the /home/oracle directory.

9. Open the pluggable database, check for pdb plug-in violations, and recompile the invalid objects.

```sql
alter pluggable database ncdb open;
alter session set container=ncdb;
select message from pdb_plug_in_violations where type like '%ERR%' and status <> 'RESOLVED';
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl -n 1 -c 'ncdb' -e -b utlrp -d $ORACLE_HOME/rdbms/admin utlrp.sql
```

10. Execute noncdb_to_pdb.sql to update the data dictionary.

```sql
sqlplus / as sysdba
alter session set container=ncdb;
@$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql;
```

Shut down and restart the container DB. The ncdb is taken out of restricted mode.

Migrate on-premises Oracle databases to Azure with PDB relocation

Oracle PDB relocation with the maximum-availability option employs PDB hot-clone technology, which allows source PDB availability while the PDB is copying over to the target. At switchover, user connections are redirected to the target PDB automatically. Thus, downtime is minimized independent of the size of the PDB. NetApp provides an Ansible-based toolkit that automates the migration procedure.
1. Create a CDB in the Azure public cloud on an Azure VM with the same version and patch level.

2. From the Ansible controller, clone a copy of the automation toolkit.

```bash
git clone https://github.com/NetApp-Automation/na_ora_aws_migration.git
```

3. Read the instruction in the README file.

4. Configure the Ansible host variable files for both the source and target Oracle servers and the DB server host's configuration file for name resolution.

5. Install the Ansible controller prerequisites on Ansible controller.

```bash
ansible-playbook -i hosts requirements.yml
ansible-galaxy collection install -r collections/requirements.yml --force
```

6. Execute any pre-migration tasks against the on-premises server.

```bash
ansible-playbook -i hosts ora_pdb_relocate.yml -u admin -k -K -t ora_pdb_relo_onprem
```

   The admin user is the management user on the on-premises Oracle server host with sudo privileges. The admin user is authenticated with a password.

7. Execute Oracle PDB relocation from on-premises to the target Azure Oracle host.

```bash
ansible-playbook -i hosts ora_pdb_relocate.yml -u azureuser --private -key db1.pem -t ora_pdb_relo_primary
```

   The Ansible controller can be located either on-premises or in the Azure cloud. The controller needs connectivity to the on-premises Oracle server host and the Azure Oracle VM host. The Oracle database port (such as 1521) is open between the on-premises Oracle server host and the Azure Oracle VM host.

**Additional Oracle database migration options**

Please see the Microsoft documentation for additional migration options: [Oracle database migration decision process](#).

**NVA-1155: Oracle 19c RAC databases on FlexPod Datacenter with Cisco UCS and NetApp AFF A800 over FC - Design and deployment guide**

Allen Cao, NetApp

This design and deployment guide for Oracle 19c RAC databases on FlexPod Datacenter with Cisco UCS and
NetApp AFF A800 over FC provides details of the solution design as well as step-by-step deployment processes for hosting Oracle RAC databases on most recent FlexPod Datacenter infrastructure with the Oracle Linux 8.2 operating system and a Red Hat compatible kernel.

NVA-1155: Oracle 19c RAC databases on FlexPod Datacenter with Cisco UCS and NetApp AFF A800 over FC

TR-4250: SAP with Oracle on UNIX and NFS with NetApp Clustered Data ONTAP and SnapManager for SAP 3.4

Nils Bauer, NetApp

TR-4250 addresses the challenges of designing storage solutions to support SAP business suite products using an Oracle database. The primary focus of this document is the common storage infrastructure design, deployment, operation, and management challenges faced by business and IT leaders who use the latest generation of SAP solutions. The recommendations in this document are generic; they are not specific to an SAP application or to the size and scope of the SAP implementation. TR-4250 assumes that the reader has a basic understanding of the technology and operation of NetApp and SAP products. TR-4250 was developed based on the interaction of technical staff from NetApp, SAP, Oracle, and our customers.

TR-4250: SAP with Oracle on UNIX and NFS with NetApp Clustered Data ONTAP and SnapManager for SAP 3.4

Deploying Oracle Database

Solution Overview

Automated Deployment of Oracle19c for ONTAP on NFS

Organizations are automating their environments to gain efficiencies, accelerate deployments, and reduce manual effort. Configuration management tools like Ansible are being used to streamline enterprise database operations. In this solution, we demonstrate how you can use Ansible to automate the provisioning and configuration of Oracle 19c with NetApp ONTAP. By enabling storage administrators, systems administrators, and DBAs to consistently and rapidly deploy new storage, configure database servers, and install Oracle 19c software, you achieve the following benefits:

- Eliminate design complexities and human errors, and implement a repeatable consistent deployment and best practices
- Decrease time for provisioning of storage, configuration of DB hosts, and Oracle installation
- Increase database administrators, systems and storage administrators productivity
- Enable scaling of storage and databases with ease

NetApp provides customers with validated Ansible modules and roles to accelerate deployment, configuration, and lifecycle management of your Oracle database environment. This solution provides instruction and Ansible playbook code, to help you:

- Create and configure ONTAP NFS storage for Oracle Database
- Install Oracle 19c on RedHat Enterprise Linux 7/8 or Oracle Linux 7/8
- Configure Oracle 19c on ONTAP NFS storage

For more details or to begin, please see the overview videos below.
This solution has been designed to be run in an AWX/Tower environment or by CLI on an Ansible control host.

**AWX/Tower**

For AWX/Tower environments, you are guided through creating an inventory of your ONTAP cluster management and Oracle server (IPs and hostnames), creating credentials, configuring a project that pulls the Ansible code from NetApp Automation Github, and the Job Template that launches the automation.

1. Fill out the variables specific to your environment, and copy and paste them into the Extra Vars fields in your job template.
2. After the extra vars have been added to your job template, you can launch the automation.
3. The job template is run in three phases by specifying tags for ontap_config, linux_config, and oracle_config.

**CLI via the Ansible control host**

1. To configure the Linux host so that is can be used as an Ansible control host [click here for detailed instructions]
2. After the Ansible control host is configured, you can git clone the Ansible Automation repository.
3. Edit the hosts file with the IPs and/or hostnames of your ONTAP cluster management and Oracle server’s management IPs.
4. Fill out the variables specific to your environment, and copy and paste them into the vars.yml file.
5. Each Oracle host has a variable file identified by its hostname that contains host-specific variables.
6. After all variable files have been completed, you can run the playbook in three phases by specifying tags for ontap_config, linux_config, and oracle_config.
### Requirements

<table>
<thead>
<tr>
<th>Environment</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ansible environment</td>
<td>AWX/Tower or Linux host to be the Ansible control host</td>
</tr>
<tr>
<td></td>
<td>Ansible v.2.10 and higher</td>
</tr>
<tr>
<td></td>
<td>Python 3</td>
</tr>
<tr>
<td></td>
<td>Python libraries</td>
</tr>
<tr>
<td></td>
<td>- netapp-lib</td>
</tr>
<tr>
<td></td>
<td>- xmltodict</td>
</tr>
<tr>
<td></td>
<td>- jmespath</td>
</tr>
<tr>
<td>ONTAP</td>
<td>ONTAP version 9.3 - 9.7</td>
</tr>
<tr>
<td></td>
<td>Two data aggregates</td>
</tr>
<tr>
<td></td>
<td>NFS vlan and ifgrp created</td>
</tr>
<tr>
<td>Oracle server(s)</td>
<td>RHEL 7/8</td>
</tr>
<tr>
<td></td>
<td>Oracle Linux 7/8</td>
</tr>
<tr>
<td></td>
<td>Network interfaces for NFS, public, and optional mgmt</td>
</tr>
<tr>
<td></td>
<td>Oracle installation files on Oracle servers</td>
</tr>
</tbody>
</table>

### Automation Details

This automated deployment is designed with a single Ansible playbook that consists of three separate roles. The roles are for ONTAP, Linux, and Oracle configurations. The following table describes which tasks are being automated.

<table>
<thead>
<tr>
<th>Role</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ontap_config</td>
<td>Pre-check of the ONTAP environment</td>
</tr>
<tr>
<td></td>
<td>Creation of NFS based SVM for Oracle</td>
</tr>
<tr>
<td></td>
<td>Creation of export policy</td>
</tr>
<tr>
<td></td>
<td>Creation of volumes for Oracle</td>
</tr>
<tr>
<td></td>
<td>Creation of NFS LIFs</td>
</tr>
<tr>
<td>linux_config</td>
<td>Create mount points and mount NFS volumes</td>
</tr>
<tr>
<td></td>
<td>Verify NFS mounts</td>
</tr>
<tr>
<td></td>
<td>OS specific configuration</td>
</tr>
<tr>
<td></td>
<td>Create Oracle directories</td>
</tr>
<tr>
<td></td>
<td>Configure hugepages</td>
</tr>
<tr>
<td></td>
<td>Disable SELinux and firewall daemon</td>
</tr>
<tr>
<td></td>
<td>Enable and start chronyd service</td>
</tr>
<tr>
<td></td>
<td>increase file descriptor hard limit</td>
</tr>
<tr>
<td></td>
<td>Create pam.d session file</td>
</tr>
</tbody>
</table>
### Role and Tasks

<table>
<thead>
<tr>
<th>Role</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>oracle_config</td>
<td>Oracle software installation</td>
</tr>
<tr>
<td></td>
<td>Create Oracle listener</td>
</tr>
<tr>
<td></td>
<td>Create Oracle databases</td>
</tr>
<tr>
<td></td>
<td>Oracle environment configuration</td>
</tr>
<tr>
<td></td>
<td>Save PDB state</td>
</tr>
<tr>
<td></td>
<td>Enable instance archive mode</td>
</tr>
<tr>
<td></td>
<td>Enable DNFS client</td>
</tr>
<tr>
<td></td>
<td>Enable database auto startup and shutdown between OS reboots</td>
</tr>
</tbody>
</table>

### Default parameters

To simplify automation, we have preset many required Oracle deployment parameters with default values. It is generally not necessary to change the default parameters for most deployments. A more advanced user can make changes to the default parameters with caution. The default parameters are located in each role folder under defaults directory.

### Deployment instructions

Before starting, download the following Oracle installation and patch files and place them in the `/tmp/archive` directory with read, write, and execute access for all users on each DB server to be deployed. The automation tasks look for the named installation files in that particular directory for Oracle installation and configuration.

```
LINUX.X64_193000_db_home.zip -- 19.3 base installer
p31281355_190000_Linux-x86-64.zip -- 19.8 RU patch
p6880880_190000_Linux-x86-64.zip -- opatch version 12.2.0.1.23
```

### License

You should read license information as stated in the Github repository. By accessing, downloading, installing, or using the content in this repository, you agree the terms of the license laid out [here](#).

Note that there are certain restrictions around producing and/or sharing any derivative works with the content in this repository. Please make sure you read the terms of the License before using the content. If you do not agree to all of the terms, do not access, download, or use the content in this repository.

After you are ready, click [here](#) for detailed AWX/Tower deployment procedures or [here](#) for CLI deployment.

### Step-by-step deployment procedure

#### AWX/Tower deployment Oracle 19c Database

1. **Create the inventory, group, hosts, and credentials for your environment**

   This section describes the setup of inventory, groups, hosts, and access credentials in AWX/Ansible Tower that prepare the environment for consuming NetApp automated solutions.
1. Configure the inventory.
   a. Navigate to Resources → Inventories → Add, and click Add Inventory.
   b. Provide the name and organization details, and click Save.
   c. On the Inventories page, click the inventory created.
   d. If there are any inventory variables, paste them in the variables field.
   e. Navigate to the Groups sub-menu and click Add.
   f. Provide the name of the group for ONTAP, paste the group variables (if any) and click Save.
   g. Repeat the process for another group for Oracle.
   h. Select the ONTAP group created, go to the Hosts sub-menu and click Add New Host.
      i. Provide the IP address of the ONTAP cluster management IP, paste the host variables (if any), and click Save.
      j. This process must be repeated for the Oracle group and Oracle host(s) management IP/hostname.

2. Create credential types. For solutions involving ONTAP, you must configure the credential type to match username and password entries.
   a. Navigate to Administration → Credential Types, and click Add.
   b. Provide the name and description.
   c. Paste the following content in Input Configuration:

   ```json
   fields:
   - id: username
     type: string
     label: Username
   - id: password
     type: string
     label: Password
     secret: true
   - id: vsadmin_password
     type: string
     label: vsadmin_password
     secret: true
   ```
   a. Paste the following content into Injector Configuration:

   ```json
   extra_vars:
   password: '{{$ password }}'
   username: '{{$ username }}'
   vsadmin_password: '{{$ vsadmin_password }}'
   ```

1. Configure the credentials.
   a. Navigate to Resources → Credentials, and click Add.
   b. Enter the name and organization details for ONTAP.
c. Select the custom Credential Type you created for ONTAP.

d. Under Type Details, enter the username, password, and vsadmin_password.

e. Click Back to Credential and click Add.

f. Enter the name and organization details for Oracle.

g. Select the Machine credential type.

h. Under Type Details, enter the Username and Password for the Oracle hosts.

i. Select the correct Privilege Escalation Method, and enter the username and password.

2. Create a project

1. Go to Resources → Projects, and click Add.

   a. Enter the name and organization details.

   b. Select Git in the Source Control Credential Type field.

   c. Enter https://github.com/NetApp-Automation/na_oracle19c_deploy.git as the source control URL.

   d. Click Save.

   e. The project might need to sync occasionally when the source code changes.

3. Configure Oracle host_vars

   The variables defined in this section are applied to each individual Oracle server and database.

   1. Input your environment-specific parameters in the following embedded Oracle hosts variables or host_vars form.

   ![Host VARS Config](image)

   The items in blue must be changed to match your environment.

   Host VARS Config

```
# Add your Oracle Host
ansible_host: "10.61.180.15"

# Oracle db log archive mode: true - ARCHIVELOG or false - NOARCHIVELOG
log_archive_mode: "true"

# Number of pluggable databases per container instance identified by sid. Pdb_name specifies the prefix for container database naming in this case cdb2_pdb1, cdb2_pdb2, cdb2_pdb3
oracle_sid: "cdb2"
pdb_num: "3"
pdb_name: "{{ oracle_sid }}_pdb"
```
# CDB listener port, use different listener port for additional CDB on same host
listener_port: "1523"

# CDB is created with SGA at 75% of memory_limit, MB. Consider how many databases to be hosted on the node and how much ram to be allocated to each DB. The grand total SGA should not exceed 75% available RAM on node.
memory_limit: "5464"

# Set "em_configuration: DBEXPRESS" to install enterprise manager express and choose a unique port from 5500 to 5599 for each sid on the host. Leave them blank if em express is not installed.
em_configuration: "DBEXPRESS"
em_express_port: "5501"

# {{groups.oracle[0]}} represents first Oracle DB server as defined in Oracle hosts group [oracle]. For concurrent multiple Oracle DB servers deployment, [0] will be incremented for each additional DB server. For example, {{groups.oracle[1]}} represents DB server 2, {{groups.oracle[2]}} represents DB server 3 ... As a good practice and the default, minimum three volumes is allocated to a DB server with corresponding /u01, /u02, /u03 mount points, which store oracle binary, oracle data, and oracle recovery files respectively. Additional volumes can be added by click on "More NFS volumes" but the number of volumes allocated to a DB server must match with what is defined in global vars file by volumes_nfs parameter, which dictates how many volumes are to be created for each DB server.

host_datastores_nfs:
  - {vol_name: "{{groups.oracle[0]}}_u01", aggr_name: "aggr01_node01", lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u02", aggr_name: "aggr01_node01", lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u03", aggr_name: "aggr01_node01", lif: "172.21.94.200", size: "25"}

a. Fill in all variables in the blue fields.
b. After completing variables input, click the Copy button on the form to copy all variables to be transferred to AWX or Tower.
c. Navigate back to AWX or Tower and go to Resources → Hosts, and select and open the Oracle server configuration page.
d. Under the Details tab, click edit and paste the copied variables from step 1 to the Variables field under the YAML tab.
e. Click Save.
f. Repeat this process for any additional Oracle servers in the system.
### 4. Configure global variables

Variables defined in this section apply to all Oracle hosts, databases, and the ONTAP cluster.

1. Input your environment-specific parameters in following embedded global variables or vars form.

   The items in blue must be changed to match your environment.

```plaintext
#Inventory group name
#Default inventory group name - 'ontap'
#Change only if you are changing the group name either in inventory/hosts file or in inventory groups in case of AWX/Tower
hosts_group: "ontap"

#CA_signed_certificates (ONLY CHANGE to 'true' IF YOU ARE USING CA SIGNED CERTIFICATES)
ca_signed_certs: "false"

#Names of the Nodes in the ONTAP Cluster
nodes:
  - "AFF-01"
  - "AFF-02"

#Storage VLANs
#Add additional rows for vlans as necessary
storage_vlans:
  - {vlan_id: "203", name: "infra_NFS", protocol: "NFS"}

More Storage VLANs
Enter Storage VLANs details

#Details of the Data Aggregates that need to be created
#If Aggregate creation takes longer, subsequent tasks of creating volumes may fail.
#There should be enough disks already zeroed in the cluster, otherwise aggregate create will zero the disks and will take long time
data_aggregates:
  - {aggr_name: "aggr01_node01"}
  - {aggr_name: "aggr01_node02"}
```
# SVM name
svm_name: "ora_svm"

# SVM Management LIF Details
svm_mgmt_details:
  - {address: "172.21.91.100", netmask: "255.255.255.0", home_port: "e0M"}

# NFS storage parameters when data_protocol set to NFS. Volume named after Oracle hosts name identified by mount point as follow for oracle DB server 1. Each mount point dedicates to a particular Oracle files: u01 - Oracle binary, u02 - Oracle data, u03 - Oracle redo. Add additional volumes by click on "More NFS volumes" and also add the volumes list to corresponding host_vars as host_datastores_nfs variable. For multiple DB server deployment, additional volumes sets needs to be added for additional DB server. Input variable "{{groups.oracle[1]}}_u01", "{{groups.oracle[1]}}_u02", and "{{groups.oracle[1]}}_u03" as vol_name for second DB server. Place volumes for multiple DB servers alternatingly between controllers for balanced IO performance, e.g. DB server 1 on controller node1, DB server 2 on controller node2 etc. Make sure match lif address with controller node.

volumes_nfs:
  - {vol_name: "{{groups.oracle[0]}}_u01", aggr_name: "aggr01_node01", lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u02", aggr_name: "aggr01_node01", lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u03", aggr_name: "aggr01_node01", lif: "172.21.94.200", size: "25"}

# NFS LIFs IP address and netmask
nfs_lifs_details:
  - address: "172.21.94.200" # for node-1
    netmask: "255.255.255.0"
  - address: "172.21.94.201" # for node-2
    netmask: "255.255.255.0"

# NFS client match
client_match: "172.21.94.0/24"

#################################################################
### Linux env specific config variables ###
#################################################################

#NFS Mount points for Oracle DB volumes
mount_points:
  - "/u01"
  - "/u02"
  - "/u03"

# Up to 75% of node memory size divided by 2mb. Consider how many databases to be hosted on the node and how much ram to be allocated to each DB.
# Leave it blank if hugepage is not configured on the host.

hugepages_nr: "1234"

# RedHat subscription username and password

redhat_sub_username: "xxx"
redhat_sub_password: "xxx"

####################################################
### DB env specific install and config variables ###
####################################################

db_domain: "your.domain.com"

# Set initial password for all required Oracle passwords. Change them after installation.

initial_pwd_all: "netapp123"

1. Fill in all variables in blue fields.
2. After completing variables input, click the Copy button on the form to copy all variables to be transferred to AWX or Tower into the following job template.

5. Configure and launch the job template.

1. Create the job template.
   a. Navigate to Resources → Templates → Add and click Add Job Template.
   b. Enter the name and description
   c. Select the Job type; Run configures the system based on a playbook, and Check performs a dry run of a playbook without actually configuring the system.
   d. Select the corresponding inventory, project, playbook, and credentials for the playbook.
   e. Select the all_playbook.yml as the default playbook to be executed.
   f. Paste global variables copied from step 4 into the Template Variables field under the YAML tab.
   g. Check the box Prompt on Launch in the Job Tags field.
   h. Click Save.
2. Launch the job template.
   a. Navigate to Resources → Templates.
   b. Click the desired template and then click Launch.
   c. When prompted on launch for Job Tags, type in requirements_config. You might need to click the Create Job Tag line below requirements_config to enter the job tag.

   requirements_config ensures that you have the correct libraries to run the other roles.

   a. Click Next and then Launch to start the job.
   b. Click View → Jobs to monitor the job output and progress.
   c. When prompted on launch for Job Tags, type in ontap_config. You might need to click the Create "Job Tag" line right below ontap_config to enter the job tag.
   d. Click Next and then Launch to start the job.
   e. Click View → Jobs to monitor the job output and progress
   f. After the ontap_config role has completed, run the process again for linux_config.
   g. Navigate to Resources → Templates.
   h. Select the desired template and then click Launch.
   i. When prompted on launch for the Job Tags type in linux_config, you might need to select the Create "job tag" line right below linux_config to enter the job tag.
   j. Click Next and then Launch to start the job.
   k. Select View → Jobs to monitor the job output and progress.
   l. After the linux_config role has completed, run the process again for oracle_config.
   m. Go to Resources → Templates.
   n. Select the desired template and then click Launch.
   o. When prompted on launch for Job Tags, type oracle_config. You might need to select the Create "Job Tag" line right below oracle_config to enter the job tag.
   p. Click Next and then Launch to start the job.
   q. Select View → Jobs to monitor the job output and progress.

6. Deploy additional database on same Oracle host

The Oracle portion of the playbook creates a single Oracle container database on an Oracle server per execution. To create additional container databases on the same server, complete the following steps.

1. Revise host_vars variables.
   a. Go back to step 2 - Configure Oracle host_vars.
   b. Change the Oracle SID to a different naming string.
   c. Change the listener port to different number.
   d. Change the EM Express port to a different number if you are installing EM Express.
   e. Copy and paste the revised host variables to the Oracle Host Variables field in the Host Configuration Detail tab.
2. Launch the deployment job template with only the oracle_config tag.
3. Log in to Oracle server as oracle user and execute the following commands:

```
ps -ef | grep ora
```

This will list oracle processes if installation completed as expected and oracle DB started

4. Log in to the database to check the db configuration settings and the PDBs created with the following command sets.
[oracle@localhost ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Thu May 6 12:52:51 2021
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL>

SQL> select name, log_mode from v$database;
NAME      LOG_MODE
--------- ------------
CDB2      ARCHIVELOG

SQL> show pdbs

CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
2 PDB$SEED                       READ ONLY  NO
3 CDB2_PDB1                      READ WRITE NO
4 CDB2_PDB2                      READ WRITE NO
5 CDB2_PDB3                      READ WRITE NO

col svrname form a30
col dirname form a30
select svrname, dirname, nfsversion from v$dnfs_servers;

SVRNAME                        DIRNAME                        NFSVERSION
------------------------------ ------------------------------ ------------
----------------
172.21.126.200                 /rhelora03_u02                 NFSv3.0
172.21.126.200                 /rhelora03_u03                 NFSv3.0
172.21.126.200                 /rhelora03_u01                 NFSv3.0

This confirms that dNFS is working properly.

5. Connect to database via listener to check the Oracle listener configuration with the following command. Change to the appropriate listener port and database service name.
This confirms that Oracle listener is working properly.

Where to go for help?

If you need help with the toolkit, please join the NetApp Solution Automation community support slack channel and look for the solution-automation channel to post your questions or inquires.

Step-by-step deployment procedure

This document details the deployment of Oracle 19c using the automation command line interface (cli).

CLI deployment Oracle 19c Database

This section covers the steps required to prepare and deploy Oracle19c Database with the CLI. Make sure that you have reviewed the Getting Started and Requirements section and prepared your environment accordingly.

Download Oracle19c repo

1. From your ansible controller, run the following command:

   ```bash
git clone https://github.com/NetApp-Automation/na_oracle19c_deploy.git
   ```

2. After downloading the repository, change directories to na_oracle19c_deploy <cd na_oracle19c_deploy>.
Edit the hosts file

Complete the following before deployment:

1. Edit your hosts file na_oracle19c_deploy directory.
2. Under [ontap], change the IP address to your cluster management IP.
3. Under the [oracle] group, add the oracle hosts names. The host name must be resolved to its IP address either through DNS or the hosts file, or it must be specified in the host.
4. After you have completed these steps, save any changes.

The following example depicts a host file:

```plaintext
#ONTAP Host
[ontap]
"10.61.184.183"

#Oracle hosts
[oracle]
"rtpora01"
"rtpora02"
```

This example executes the playbook and deploys oracle 19c on two oracle DB servers concurrently. You can also test with just one DB server. In that case, you only need to configure one host variable file.

- The playbook executes the same way regardless of how many Oracle hosts and databases you deploy.

Edit the host_name.yml file under host_vars

Each Oracle host has its host variable file identified by its host name that contains host-specific variables. You can specify any name for your host. Edit and copy the host_vars from the Host VARS Config section and paste it into your desired host_name.yml file.

- The items in blue must be changed to match your environment.

Host VARS Config

```plaintext
# Add your Oracle Host
```

1152
ansible_host: "10.61.180.15"

# Oracle db log archive mode: true - ARCHIVELOG or false - NOARCHIVELOG
log_archive_mode: "true"

# Number of pluggable databases per container instance identified by sid.
Pdb_name specifies the prefix for container database naming in this case
cdb2_pdb1, cdb2_pdb2, cdb2_pdb3
oracle_sid: "cdb2"
pdb_num: "3"
pdb_name: "{{ oracle_sid }}_pdb"

# CDB listener port, use different listener port for additional CDB on same host
listener_port: "1523"

# CDB is created with SGA at 75% of memory_limit, MB. Consider how many databases to be hosted on the node and how much ram to be allocated to each DB. The grand total SGA should not exceed 75% available RAM on node.
memory_limit: "5464"

# Set "em_configuration: DBEXPRESS" to install enterprise manager express and choose a unique port from 5500 to 5599 for each sid on the host.
# Leave them black if em express is not installed.
em_configuration: "DBEXPRESS"
em_express_port: "5501"

# {{groups.oracle[0]}} represents first Oracle DB server as defined in Oracle hosts group [oracle]. For concurrent multiple Oracle DB servers deployment, [0] will be incremented for each additional DB server. For example, {{groups.oracle[1]}}" represents DB server 2,
"{{groups.oracle[2]}}" represents DB server 3 ... As a good practice and the default, minimum three volumes is allocated to a DB server with corresponding /u01, /u02, /u03 mount points, which store oracle binary, oracle data, and oracle recovery files respectively. Additional volumes can be added by click on "More NFS volumes" but the number of volumes allocated to a DB server must match with what is defined in global vars file by volumes_nfs parameter, which dictates how many volumes are to be created for each DB server.
host_datastores_nfs:
  - {vol_name: "{{groups.oracle[0]}}_u01", aggr_name: "aggr01_node01",
    lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u02", aggr_name: "aggr01_node01",
    lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u03", aggr_name: "aggr01_node01",
    lif: "172.21.94.200", size: "25"}
Edit the vars.yml file

The vars.yml file consolidates all environment-specific variables (ONTAP, Linux, or Oracle) for Oracle deployment.

1. Edit and copy the variables from the VARS section and paste these variables into your vars.yml file.

---

```yaml
# Inventory group name
hosts_group: "ontap"

# CA signed certificates (ONLY CHANGE to 'true' IF YOU ARE USING CA SIGNED CERTIFICATES)
ca_signed_certs: "false"

# Names of the Nodes in the ONTAP Cluster
nodes:
  - "AFF-01"
  - "AFF-02"

# Storage VLANs
storage_vlans:
  - {vlan_id: "203", name: "infra_NFS", protocol: "NFS"}

# Details of the Data Aggregates that need to be created
# If Aggregate creation takes longer, subsequent tasks of creating volumes may fail.
# There should be enough disks already zeroed in the cluster, otherwise aggregate create will zero the disks and will take long time
data_aggregates:
  - {aggr_name: "aggr01_node01"}
  - {aggr_name: "aggr01_node02"}

# SVM name
```
---
svm_name: "ora_svm"

# SVM Management LIF Details
svm_mgmt_details:
  - {address: "172.21.91.100", netmask: "255.255.255.0", home_port: "e0M"}

# NFS storage parameters when data_protocol set to NFS. Volume named after Oracle hosts name identified by mount point as follow for oracle DB server 1. Each mount point dedicates to a particular Oracle files: u01 - Oracle binary, u02 - Oracle data, u03 - Oracle redo. Add additional volumes by click on "More NFS volumes" and also add the volumes list to corresponding host_vars as host_datastores_nfs variable. For multiple DB server deployment, additional volumes sets needs to be added for additional DB server. Input variable "{{groups.oracle[1]}}_u01", "{{groups.oracle[1]}}_u02", and "{{groups.oracle[1]}}_u03" as vol_name for second DB server. Place volumes for multiple DB servers alternatingly between controllers for balanced IO performance, e.g. DB server 1 on controller node1, DB server 2 on controller node2 etc. Make sure match lif address with controller node.

volumes_nfs:
  - {vol_name: "{{groups.oracle[0]}}_u01", aggr_name: "aggr01_node01", lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u02", aggr_name: "aggr01_node01", lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u03", aggr_name: "aggr01_node01", lif: "172.21.94.200", size: "25"}

#NFS LIFs IP address and netmask

nfs_lifs_details:
  - address: "172.21.94.200" #for node-1
    netmask: "255.255.255.0"
  - address: "172.21.94.201" #for node-2
    netmask: "255.255.255.0"

#NFS client match

client_match: "172.21.94.0/24"

#NFS Mount points for Oracle DB volumes

mount_points:
- "/u01"
- "/u02"
- "/u03"

# Up to 75% of node memory size divided by 2mb. Consider how many
databases to be hosted on the node and how much ram to be allocated to
each DB.
# Leave it blank if hugepage is not configured on the host.

hugepages_nr: "1234"

# RedHat subscription username and password

redhat_sub_username: "xxx"
redhat_sub_password: "xxx"

####################################################
### DB env specific install and config variables ###
####################################################

db_domain: "your.domain.com"

# Set initial password for all required Oracle passwords. Change them
after installation.

initial_pwd_all: "netappl23"

Run the playbook

After completing the required environment prerequisites and copying the variables into vars.yml and
your_host.yml, you are now ready to deploy the playbooks.

1. Run the ONTAP playbook by passing the correct tags and ONTAP cluster username. Fill the password for
ONTAP cluster, and vsadmin when prompted.

   ansible-playbook -i hosts all_playbook.yml -u username -k -K -t
   ontap_config -e @vars/vars.yml

2. Run the Linux playbook to execute Linux portion of deployment. Input for admin ssh password as well as
sudo password.

   ansible-playbook -i hosts all_playbook.yml -u username -k -K -t
   linux_config -e @vars/vars.yml
3. Run the Oracle playbook to execute Oracle portion of deployment. Input for admin ssh password as well as sudo password.

```
ansible-playbook -i hosts all_playbook.yml -u username -k -K -t oracle_config -e @vars/vars.yml
```

**Deploy Additional Database on Same Oracle Host**

The Oracle portion of the playbook creates a single Oracle container database on an Oracle server per execution. To create additional container database on the same server, complete the following steps:

1. Revise the host_vars variables.
   a. Go back to step 3 - Edit the `host_name.yml` file under `host_vars`.
   b. Change the Oracle SID to a different naming string.
   c. Change the listener port to different number.
   d. Change the EM Express port to a different number if you have installed EM Express.
   e. Copy and paste the revised host variables to the Oracle host variable file under `host_vars`.

2. Execute the playbook with the `oracle_config` tag as shown above in Run the playbook.

**Validate Oracle installation**

1. Log in to Oracle server as oracle user and execute the following commands:

```
ps -ef | grep ora
```

This will list oracle processes if installation completed as expected and oracle DB started

2. Log in to the database to check the db configuration settings and the PDBs created with the following command sets.
[oracle@localhost ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Thu May 6 12:52:51 2021
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL>

SQL> select name, log_mode from v$database;
NAME      LOG_MODE
--------- ------------
CDB2      ARCHIVELOG

SQL> show pdbs

CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
2 PDB$SEED                       READ ONLY  NO
3 CDB2_PDB1                      READ WRITE NO
4 CDB2_PDB2                      READ WRITE NO
5 CDB2_PDB3                      READ WRITE NO

col svrname form a30
col dirname form a30
select svrname, dirname, nfsversion from v$dnfs_servers;

SVRNAME                        DIRNAME                        NFSVERSION
------------------------------ ------------------------------ ------------------
172.21.126.200                 /rhelora03_u02                 NFSv3.0
172.21.126.200                 /rhelora03_u03                 NFSv3.0
172.21.126.200                 /rhelora03_u01                 NFSv3.0

This confirms that dNFS is working properly.

3. Connect to database via listener to check the Oracle listener configuration with the following command.
   Change to the appropriate listener port and database service name.
[oracle@localhost ~]$ sqlplus
system@//localhost:1523/cdb2_pdb1.cie.netapp.com

SQL*Plus: Release 19.0.0.0.0 - Production on Thu May 6 13:19:57 2021
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Enter password:
Last Successful login time: Wed May 05 2021 17:11:11 -04:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL> show user
USER is "SYSTEM"
SQL> show con_name
CON_NAME
CDB2_PDB1

This confirms that Oracle listener is working properly.

Where to go for help?

If you need help with the toolkit, please join the NetApp Solution Automation community support slack channel and look for the solution-automation channel to post your questions or inquires.

Solution Overview

Automated Data Protection for Oracle Databases

Organizations are automating their environments to gain efficiencies, accelerate deployments, and reduce manual effort. Configuration management tools like Ansible are being used to streamline enterprise database operations. In this solution, we demonstrate how you can use Ansible to automate the data protection of Oracle with NetApp ONTAP. By enabling storage administrators, systems administrators, and DBAs to consistently and rapidly setup data replication to an offsite data center or to public cloud, you achieve the following benefits:

- Eliminate design complexities and human errors, and implement a repeatable consistent deployment and best practices
- Decrease time for configuration of Intercluster replication, CVO instantiation, and recovery of Oracle databases
- Increase database administrators, systems and storage administrators productivity
- Provides database recovery workflow for ease of testing a DR scenario.

NetApp provides customers with validated Ansible modules and roles to accelerate deployment, configuration, and lifecycle management of your Oracle database environment. This solution provides instruction and Ansible
playbook code, to help you:

On Prem to on prem replication

• Create intercluster lifs on source and destination
• Establish cluster and vserver peering
• Create and initialize SnapMirror of Oracle volumes
• Create a replication schedule through AWX/Tower for Oracle binaries, databases, and logs
• Restore Oracle DB on the destination, and bring database online

On Prem to CVO in AWS

• Create AWS connector
• Create CVO instance in AWS
• Add On-Prem cluster to Cloud Manager
• Create intercluster lifs on source
• Establish cluster and vserver peering
• Create and initialize SnapMirror of Oracle volumes
• Create a replication schedule through AWX/Tower for Oracle binaries, databases, and logs
• Restore Oracle DB on the destination, and bring database online

After you are ready, click here for getting started with the solution.

Getting started

This solution has been designed to be run in an AWX/Tower environment.

AWX/Tower

For AWX/Tower environments, you are guided through creating an inventory of your ONTAP cluster management and Oracle server (IPs and hostnames), creating credentials, configuring a project that pulls the Ansible code from NetApp Automation Github, and the Job Template that launches the automation.

1. The solution has been designed to run in a private cloud scenario (on-premise to on-premise), and hybrid cloud (on-premise to public cloud Cloud Volumes ONTAP [CVO])
2. Fill out the variables specific to your environment, and copy and paste them into the Extra Vars fields in your job template.
3. After the extra vars have been added to your job template, you can launch the automation.
4. The automation is set to be ran three phases (Setup, Replication Schedule for Oracle Binaries, Database, Logs, and Replication Schedule just for Logs), and a forth phase to recovering the database at a DR site.
5. For detailed instructions for obtaining the keys and tokens necessary for the CVO Data Protection visit Gather Pre-requisites For CVO and Connector Deployments

Requirements
<table>
<thead>
<tr>
<th>Environment</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ansible environment</td>
<td>AWX/Tower</td>
</tr>
<tr>
<td></td>
<td>Ansible v.2.10 and higher</td>
</tr>
<tr>
<td></td>
<td>Python 3</td>
</tr>
<tr>
<td></td>
<td>Python libraries</td>
</tr>
<tr>
<td></td>
<td>- netapp-lib</td>
</tr>
<tr>
<td></td>
<td>- xmltodict</td>
</tr>
<tr>
<td></td>
<td>- jmespath</td>
</tr>
<tr>
<td>ONTAP</td>
<td>ONTAP version 9.8 +</td>
</tr>
<tr>
<td></td>
<td>Two data aggregates</td>
</tr>
<tr>
<td></td>
<td>NFS vlan and ifgrp created</td>
</tr>
<tr>
<td>Oracle server(s)</td>
<td>RHEL 7/8</td>
</tr>
<tr>
<td></td>
<td>Oracle Linux 7/8</td>
</tr>
<tr>
<td></td>
<td>Network interfaces for NFS, public, and optional mgmt</td>
</tr>
<tr>
<td></td>
<td>Existing Oracle environment on source, and the equivalent Linux operating system at the destination (DR Site or Public Cloud)</td>
</tr>
</tbody>
</table>

**CVO**

<table>
<thead>
<tr>
<th>Environment</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ansible environment</td>
<td>AWX/Tower</td>
</tr>
<tr>
<td></td>
<td>Ansible v.2.10 and higher</td>
</tr>
<tr>
<td></td>
<td>Python 3</td>
</tr>
<tr>
<td></td>
<td>Python libraries</td>
</tr>
<tr>
<td></td>
<td>- netapp-lib</td>
</tr>
<tr>
<td></td>
<td>- xmltodict</td>
</tr>
<tr>
<td></td>
<td>- jmespath</td>
</tr>
<tr>
<td>ONTAP</td>
<td>ONTAP version 9.8 +</td>
</tr>
<tr>
<td></td>
<td>Two data aggregates</td>
</tr>
<tr>
<td></td>
<td>NFS vlan and ifgrp created</td>
</tr>
<tr>
<td>Oracle server(s)</td>
<td>RHEL 7/8</td>
</tr>
<tr>
<td></td>
<td>Oracle Linux 7/8</td>
</tr>
<tr>
<td></td>
<td>Network interfaces for NFS, public, and optional mgmt</td>
</tr>
<tr>
<td></td>
<td>Existing Oracle environment on source, and the equivalent Linux operating system at the destination (DR Site or Public Cloud)</td>
</tr>
<tr>
<td></td>
<td>Set appropriate swap space on the Oracle EC2 instance, by default some EC2 instances are deployed with 0 swap</td>
</tr>
<tr>
<td>Environment</td>
<td>Requirements</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Cloud Manager/AWS</td>
<td>AWS Access/Secret Key</td>
</tr>
<tr>
<td></td>
<td>NetApp Cloud Manager Account</td>
</tr>
<tr>
<td></td>
<td>NetApp Cloud Manager Refresh Token</td>
</tr>
</tbody>
</table>

Automation Details
On-Prem

This automated deployment is designed with a single Ansible playbook that consists of three separate roles. The roles are for ONTAP, Linux, and Oracle configurations.

The following table describes which tasks are being automated.

<table>
<thead>
<tr>
<th>Playbook</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ontap_setup</strong></td>
<td>Pre-check of the ONTAP environment</td>
</tr>
<tr>
<td></td>
<td>Creation of Intercluster LIFs on source cluster (OPTIONAL)</td>
</tr>
<tr>
<td></td>
<td>Creation of Intercluster LIFs on destination cluster (OPTIONAL)</td>
</tr>
<tr>
<td></td>
<td>Creation of Cluster and SVM Peering</td>
</tr>
<tr>
<td></td>
<td>Creation of destination SnapMirror and Initialization of designated Oracle volumes</td>
</tr>
<tr>
<td><strong>ora_replication_cg</strong></td>
<td>Enable backup mode for each database in /etc/oratab</td>
</tr>
<tr>
<td></td>
<td>Snapshot taken of Oracle Binary and Database volumes</td>
</tr>
<tr>
<td></td>
<td>Snapmirror Updated</td>
</tr>
<tr>
<td></td>
<td>Turn off backup mode for each database in /etc/oratab</td>
</tr>
<tr>
<td><strong>ora_replication_log</strong></td>
<td>Switch current log for each database in /etc/oratab</td>
</tr>
<tr>
<td></td>
<td>Snapshot taken of Oracle Log volume</td>
</tr>
<tr>
<td></td>
<td>Snapmirror Updated</td>
</tr>
<tr>
<td><strong>ora_recovery</strong></td>
<td>Break SnapMirror</td>
</tr>
<tr>
<td></td>
<td>Enable NFS and create junction path for Oracle volumes on the destination</td>
</tr>
<tr>
<td></td>
<td>Configure DR Oracle Host</td>
</tr>
<tr>
<td></td>
<td>Mount and verify Oracle volumes</td>
</tr>
<tr>
<td></td>
<td>Recover and start Oracle database</td>
</tr>
</tbody>
</table>

CVO

This automated deployment is designed with a single Ansible playbook that consists of three separate roles. The roles are for ONTAP, Linux, and Oracle configurations.

The following table describes which tasks are being automated.
<table>
<thead>
<tr>
<th>Playbook</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>cvo_setup</td>
<td>Pre-check of the environment</td>
</tr>
<tr>
<td></td>
<td>AWS Configure/AWS Access Key ID/Secret Key/Default Region</td>
</tr>
<tr>
<td></td>
<td>Creation of AWS Role</td>
</tr>
<tr>
<td></td>
<td>Creation of NetApp Cloud Manager Connector instance in AWS</td>
</tr>
<tr>
<td></td>
<td>Creation of Cloud Volumes ONTAP (CVO) instance in AWS</td>
</tr>
<tr>
<td></td>
<td>Add On-Prem Source ONTAP Cluster to NetApp Cloud Manager</td>
</tr>
<tr>
<td></td>
<td>Creation of destination SnapMirror and Initialization of designated Oracle volumes</td>
</tr>
<tr>
<td>ora_replication_cg</td>
<td>Enable backup mode for each database in /etc/oratab</td>
</tr>
<tr>
<td></td>
<td>Snapshot taken of Oracle Binary and Database volumes</td>
</tr>
<tr>
<td></td>
<td>Snapmirror Updated</td>
</tr>
<tr>
<td></td>
<td>Turn off backup mode for each database in /etc/oratab</td>
</tr>
<tr>
<td>ora_replication_log</td>
<td>Switch current log for each database in /etc/oratab</td>
</tr>
<tr>
<td></td>
<td>Snapshot taken of Oracle Log volume</td>
</tr>
<tr>
<td></td>
<td>Snapmirror Updated</td>
</tr>
<tr>
<td>ora_recovery</td>
<td>Break SnapMirror</td>
</tr>
<tr>
<td></td>
<td>Enable NFS and create junction path for Oracle volumes on the destination CVO</td>
</tr>
<tr>
<td></td>
<td>Configure DR Oracle Host</td>
</tr>
<tr>
<td></td>
<td>Mount and verify Oracle volumes</td>
</tr>
<tr>
<td></td>
<td>Recover and start Oracle database</td>
</tr>
</tbody>
</table>

**Default parameters**

To simplify automation, we have preset many required Oracle parameters with default values. It is generally not necessary to change the default parameters for most deployments. A more advanced user can make changes to the default parameters with caution. The default parameters are located in each role folder under defaults directory.

**License**

You should read license information as stated in the Github repository. By accessing, downloading, installing, or using the content in this repository, you agree the terms of the license laid out [here](#).

Note that there are certain restrictions around producing and/or sharing any derivative works with the content in this repository. Please make sure you read the terms of the License before using the content. If you do not agree to all of the terms, do not access, download, or use the content in this repository.

After you are ready, click [here](#) for detailed AWX/Tower procedures.
Step-by-step deployment procedure

AWX/Tower Oracle Data Protection

Create the inventory, group, hosts, and credentials for your environment

This section describes the setup of inventory, groups, hosts, and access credentials in AWX/Ansible Tower that prepare the environment for consuming NetApp automated solutions.

1. Configure the inventory.
   a. Navigate to Resources → Inventories → Add, and click Add Inventory.
   b. Provide the name and organization details, and click Save.
   c. On the Inventories page, click the inventory created.
   d. Navigate to the Groups sub-menu and click Add.
   e. Provide the name oracle for your first group and click Save.
   f. Repeat the process for a second group called dr_oracle.
   g. Select the oracle group created, go to the Hosts sub-menu and click Add New Host.
   h. Provide the IP address of the Source Oracle host's management IP, and click Save.
   i. This process must be repeated for the dr_oracle group and add the DR/Destination Oracle host's management IP/hostname.

   Below are instructions for creating the credential types and credentials for either On-Prem with ONTAP, or CVO on AWS.
On-Prem

1. Configure the credentials.

2. Create Credential Types. For solutions involving ONTAP, you must configure the credential type to match username and password entries.
   a. Navigate to Administration → Credential Types, and click Add.
   b. Provide the name and description.
   c. Paste the following content in Input Configuration:

   ```
   fields:
   - id: dst_cluster_username
     type: string
     label: Destination Cluster Username
   - id: dst_cluster_password
     type: string
     label: Destination Cluster Password
     secret: true
   - id: src_cluster_username
     type: string
     label: Source Cluster Username
   - id: src_cluster_password
     type: string
     label: Source Cluster Password
     secret: true
   
   d. Paste the following content into Injector Configuration and then click Save:

   ```

   ```
   extra_vars:
   dst_cluster_username: '{{ dst_cluster_username }}'
   dst_cluster_password: '{{ dst_cluster_password }}'
   src_cluster_username: '{{ src_cluster_username }}'
   src_cluster_password: '{{ src_cluster_password }}'
   ```

3. Create Credential for ONTAP
   a. Navigate to Resources → Credentials, and click Add.
   b. Enter the name and organization details for the ONTAP Credentials
   c. Select the credential type that was created in the previous step.
   d. Under Type Details, enter the Username and Password for your Source and Destination Clusters.
   e. Click Save

4. Create Credential for Oracle
   a. Navigate to Resources → Credentials, and click Add.
   b. Enter the name and organization details for Oracle
c. Select the Machine credential type.
d. Under Type Details, enter the Username and Password for the Oracle hosts.
e. Select the correct Privilege Escalation Method, and enter the username and password.
f. Click Save
g. Repeat process if needed for a different credential for the dr_oracle host.

CVO
1. Configure the credentials.
2. Create credential types. For solutions involving ONTAP, you must configure the credential type to match username and password entries, we will also add entries for Cloud Central and AWS.
   a. Navigate to Administration → Credential Types, and click Add.
   b. Provide the name and description.
   c. Paste the following content in Input Configuration:
fields:
- id: dst_cluster_username
  type: string
  label: CVO Username
- id: dst_cluster_password
  type: string
  label: CVO Password
  secret: true
- id: cvo_svm_password
  type: string
  label: CVO SVM Password
  secret: true
- id: src_cluster_username
  type: string
  label: Source Cluster Username
- id: src_cluster_password
  type: string
  label: Source Cluster Password
  secret: true
- id: regular_id
  type: string
  label: Cloud Central ID
  secret: true
- id: email_id
  type: string
  label: Cloud Manager Email
  secret: true
- id: cm_password
  type: string
  label: Cloud Manager Password
  secret: true
- id: access_key
  type: string
  label: AWS Access Key
  secret: true
- id: secret_key
  type: string
  label: AWS Secret Key
  secret: true
- id: token
  type: string
  label: Cloud Central Refresh Token
  secret: true

d. Paste the following content into Injector Configuration and click Save:
extra_vars:
  dst_cluster_username: '{{ dst_cluster_username }}'
  dst_cluster_password: '{{ dst_cluster_password }}'
  cvo_svm_password: '{{ cvo_svm_password }}'
  src_cluster_username: '{{ src_cluster_username }}'
  src_cluster_password: '{{ src_cluster_password }}'
  regular_id: '{{ regular_id }}'
  email_id: '{{ email_id }}'
  cm_password: '{{ cm_password }}'
  access_key: '{{ access_key }}'
  secret_key: '{{ secret_key }}'
  token: '{{ token }}'

3. Create Credential for ONTAP/CVO/AWS
   a. Navigate to Resources → Credentials, and click Add.
   b. Enter the name and organization details for the ONTAP Credentials
   c. Select the credential type that was created in the previous step.
   d. Under Type Details, enter the Username and Password for your Source and CVO Clusters, Cloud Central/Manager, AWS Access/Secret Key and Cloud Central Refresh Token.
   e. Click Save

4. Create Credential for Oracle (Source)
   a. Navigate to Resources → Credentials, and click Add.
   b. Enter the name and organization details for Oracle host
   c. Select the Machine credential type.
   d. Under Type Details, enter the Username and Password for the Oracle hosts.
   e. Select the correct Privilege Escalation Method, and enter the username and password.
   f. Click Save

5. Create Credential for Oracle Destination
   a. Navigate to Resources → Credentials, and click Add.
   b. Enter the name and organization details for the DR Oracle host
   c. Select the Machine credential type.
   d. Under Type Details, enter the Username (ec2-user or if you have changed it from default enter that), and the SSH Private Key
   e. Select the correct Privilege Escalation Method (sudo), and enter the username and password if needed.
   f. Click Save

Create a project

1. Go to Resources → Projects, and click Add.
a. Enter the name and organization details.

b. Select Git in the Source Control Credential Type field.

c. Enter https://github.com/NetApp-Automation/na_oracle19c_data_protection.git as the source control URL.

d. Click Save.

e. The project might need to sync occasionally when the source code changes.

**Configure global variables**

Variables defined in this section apply to all Oracle hosts, databases, and the ONTAP cluster.

1. Input your environment-specific parameters in following embedded global variables or vars form.

The items in blue must be changed to match your environment.
hosts_group: "ontap"
ca_signed_certs: "false"

src_nodes:
- "AFF-01"
- "AFF-02"

dst_nodes:
- "DR-AFF-01"
- "DR-AFF-02"

create_source_intercluster_lifs: "yes"

source_intercluster_network_port_details:
  using_dedicated_ports: "yes"
  using_ifgrp: "yes"
  using_vlans: "yes"
  failover_for_shared_individual_ports: "yes"
  ifgrp_name: "a0a"
  vlan_id: "10"
  ports:
    - "e0b"
    - "e0g"
  broadcast_domain: "NFS"
  ipspace: "Default"
  failover_group_name: "iclifs"

source_intercluster_lif_details:
- name: "icl_1"
  address: "10.0.0.1"
  netmask: "255.255.255.0"
  home_port: "a0a-10"
  node: "AFF-01"
- name: "icl_2"
  address: "10.0.0.2"
  netmask: "255.255.255.0"
  home_port: "a0a-10"
  node: "AFF-02"

create_destination_intercluster_lifs: "yes"
destination_intercluster_network_port_details:
  using_dedicated_ports: "yes"
  using_ifgrp: "yes"
  using_vlans: "yes"
  failover_for_shared_individual_ports: "yes"
  ifgrp_name: "a0a"
  vlan_id: "10"
  ports:
    - "e0b"
    - "e0g"
  broadcast_domain: "NFS"
  ipspace: "Default"
  failover_group_name: "iclifs"

destination_intercluster_lif_details:
  - name: "icl_1"
    address: "10.0.0.3"
    netmask: "255.255.255.0"
    home_port: "a0a-10"
    node: "DR-AFF-01"
  - name: "icl_2"
    address: "10.0.0.4"
    netmask: "255.255.255.0"
    home_port: "a0a-10"
    node: "DR-AFF-02"

# Variables for SnapMirror Peering
passphrase: "your-passphrase"

# Source & Destination List
dst_cluster_name: "dst-cluster-name"
dst_cluster_ip: "dst-cluster-ip"
dst_vserver: "dst-vserver"
dst_nfs_lif: "dst-nfs-lif"
src_cluster_name: "src-cluster-name"
src_cluster_ip: "src-cluster-ip"
src_vserver: "src-vserver"

# Variable for Oracle Volumes and SnapMirror Details
cg_snapshot_name_prefix: "oracle"
src_orabinary_vols:
  - "binary_vol"
src_db_vols:
  - "db_vol"
src_archivelog_vols:
  - "log_vol"
snapmirror_policy: "async_policy_oracle"

# Export Policy Details
export_policy_details:
  name: "nfs_export_policy"
  client_match: "0.0.0.0/0"
  ro_rule: "sys"
  rw_rule: "sys"

# Linux env specific config variables
mount_points:
  - "/u01"
  - "/u02"
  - "/u03"
hugepages_nr: "1234"
redhat_sub_username: "xxx"
redhat_sub_password: "xxx"

# DB env specific install and config variables
recovery_type: "scn"
control_files:
  - "/u02/oradata/CDB2/control01.ctl"
  - "/u03/orareco/CDB2/control02.ctl"

CVO

########################################################################
### Ontap env specific config variables ###
########################################################################

#Inventory group name
#Default inventory group name - "ontap"
#Change only if you are changing the group name either in inventory/hosts file or in inventory groups in case of AWX/Tower
hosts_group: "ontap"

#CA_signed_certificates (ONLY CHANGE to "true" IF YOU ARE USING CA SIGNED CERTIFICATES)
ca_signed_certs: "false"

#Names of the Nodes in the Source ONTAP Cluster
src_nodes:
  - "AFF-01"
  - "AFF-02"

#Names of the Nodes in the Destination CVO Cluster
dst_nodes:
  - "DR-AFF-01"
  - "DR-AFF-02"

# Define whether or not to create intercluster lifs on source cluster
(ONLY CHANGE to "No" IF YOU HAVE ALREADY CREATED THE INTERCLUSTER LIFS)
create_source_intercluster_lifs: "yes"

source_intercluster_network_port_details:
  using_dedicated_ports: "yes"
  using_ifgrp: "yes"
  using_vlans: "yes"
  failover_for_shared_individual_ports: "yes"
  ifgrp_name: "a0a"
  vlan_id: "10"
  ports:
    - "e0b"
    - "e0g"
  broadcast_domain: "NFS"
  ipspace: "Default"
  failover_group_name: "iclifs"

source_intercluster_lif_details:
- name: "icl_1"
  address: "10.0.0.1"
  netmask: "255.255.255.0"
  home_port: "a0a-10"
  node: "AFF-01"
- name: "icl_2"
  address: "10.0.0.2"
  netmask: "255.255.255.0"
  home_port: "a0a-10"
  node: "AFF-02"

############################################################
### CVO Deployment Variables ###
############################################################

###### Access Keys Variables ######

# Region where your CVO will be deployed.
region_deploy: "us-east-1"

########### CVO and Connector Vars ###########

# AWS Managed Policy required to give permission for IAM role creation.
aws_policy: "arn:aws:iam::1234567:policy/OCCM"

# Specify your aws role name, a new role is created if one already does not exist.
aws_role_name: "arn:aws:iam::1234567:policy/OCCM"

# Name your connector.
connector_name: "awx_connector"

# Name of the key pair generated in AWS.
key_pair: "key_pair"

# Name of the Subnet that has the range of IP addresses in your VPC.
subnet: "subnet-12345"

# ID of your AWS security group that allows access to on-prem resources.
security_group: "sg-123123123"

# You Cloud Manager Account ID.
account: "account-A23123A"

# Name of the your CVO instance
cvo_name: "test_cvo"

# ID of the VPC in AWS.
vpc: "vpc-123123123"

#######################################################################
############################
# Variables for - Add on-prem ONTAP to Connector in Cloud Manager
#######################################################################

# For Federated users, Client ID from API Authentication Section of Cloud Central to generate access token.
sso_id: "123123123123123123123"

# For regular access with username and password, please specify "pass" as the connector_access. For SSO users, use "refresh_token" as the variable.
connector_access: "pass"

#######################################################################
############################
# Variables for SnapMirror Peering
#######################################################################
passphrase: "your-passphrase"

# Source & Destination List
# Please Enter Destination Cluster Name
dst_cluster_name: "dst-cluster-name"

# Please Enter Destination Cluster (Once CVO is Created Add this Variable to all templates)
dst_cluster_ip: "dst-cluster-ip"

# Please Enter Destination SVM to create mirror relationship
dst_vserver: "dst-vserver"

# Please Enter NFS Lif for dst vserver (Once CVO is Created Add this Variable to all templates)
dst_nfs_lif: "dst-nfs-lif"

# Please Enter Source Cluster Name
src_cluster_name: "src-cluster-name"

# Please Enter Source Cluster
src_cluster_ip: "src-cluster-ip"

# Please Enter Source SVM
src_vserver: "src-vserver"

# Variable for Oracle Volumes and SnapMirror Details
# Please Enter Source Snapshot Prefix Name
cg_snapshot_name_prefix: "oracle"

# Please Enter Source Oracle Binary Volume(s)
src_orabinary_vols:
  - "binary_vol"

# Please Enter Source Database Volume(s)
src_db_vols:
  - "db_vol"

# Please Enter Source Archive Volume(s)
## src_archivelog_vols:
- "log_vol"

# Please Enter Destination Snapmirror Policy
snapmirror_policy: "async_policy_oracle"

###############################################
# Export Policy Details
###############################################
# Enter the destination export policy details (Once CVO is Created Add this Variable to all templates)

export_policy_details:
  name: "nfs_export_policy"
  client_match: "0.0.0.0/0"
  ro_rule: "sys"
  rw_rule: "sys"

###############################################
### Linux env specific config variables ###
###############################################

# NFS Mount points for Oracle DB volumes

mount_points:
- "/u01"
- "/u02"
- "/u03"

# Up to 75% of node memory size divided by 2mb. Consider how many databases to be hosted on the node and how much ram to be allocated to each DB.
# Leave it blank if hugepage is not configured on the host.

hugepages_nr: "1234"

# RedHat subscription username and password

redhat_sub_username: "xxx"
redhat_sub_password: "xxx"

###############################################
### DB env specific install and config variables ###
###############################################

# Recovery Type (leave as scn)

recovery_type: "scn"
# Oracle Control Files

control_files:
- "/u02/oradata/CDB2/control01.ctl"
- "/u03/orareco/CDB2/control02.ctl"

## Automation Playbooks

There are four separate playbooks that need to be ran.

1. Playbook for Setting up your environment, On-Prem or CVO.
2. Playbook for replicating Oracle Binaries and Databases on a schedule
3. Playbook for replicating Oracle Logs on a schedule
4. Playbook for Recovering your database on a destination host
ONTAP/CVO Setup
ONTAP and CVO Setup

Configure and launch the job template.

1. Create the job template.
   a. Navigate to Resources → Templates → Add and click Add Job Template.
   b. Enter the name ONTAP/CVO Setup
   c. Select the Job type; Run configures the system based on a playbook.
   d. Select the corresponding inventory, project, playbook, and credentials for the playbook.
   e. Select the ontap_setup.yml playbook for an On-Prem environment or select the cvo_setup.yml for replicating to a CVO instance.
   f. Paste global variables copied from step 4 into the Template Variables field under the YAML tab.
   g. Click Save.

2. Launch the job template.
   a. Navigate to Resources → Templates.
   b. Click the desired template and then click Launch.

We will use this template and copy it out for the other playbooks.

Replication For Binary and Database Volumes
Scheduling the Binary and Database Replication Playbook

Configure and launch the job template.

1. Copy the previously created job template.
   a. Navigate to Resources → Templates.
   b. Find the ONTAP/CVO Setup Template, and on the far right click on Copy Template
   c. Click Edit Template on the copied template, and change the name to Binary and Database Replication Playbook.
   d. Keep the same inventory, project, credentials for the template.
   e. Select the ora_replication_cg.yml as the playbook to be executed.
   f. The variables will remain the same, but the CVO cluster IP will need to be set in the variable dst_cluster_ip.
   g. Click Save.

2. Schedule the job template.
   a. Navigate to Resources → Templates.
   b. Click the Binary and Database Replication Playbook template and then click Schedules at the top set of options.
   c. Click Add, add Name Schedule for Binary and Database Replication, choose the Start date/time at the beginning of the hour, choose your Local time zone, and Run frequency. Run frequency will be often the SnapMirror replication will be updated.
A separate schedule will be created for the Log volume replication, so that it can be replicated on a more frequent cadence.

Replication for Log Volumes

Scheduling the Log Replication Playbook

Configure and launch the job template.

1. Copy the previously created job template.
   a. Navigate to Resources → Templates.
   b. Find the ONTAP/CVO Setup Template, and on the far right click on Copy Template
   c. Click Edit Template on the copied template, and change the name to Log Replication Playbook.
   d. Keep the same inventory, project, credentials for the template.
   e. Select the ora_replication_logs.yml as the playbook to be executed.
   f. The variables will remain the same, but the CVO cluster IP will need to be set in the variable dst_cluster_ip.
   g. Click Save.

2. Schedule the job template.
   a. Navigate to Resources → Templates.
   b. Click the Log Replication Playbook template and then click Schedules at the top set of options.
   c. Click Add, add Name Schedule for Log Replication, choose the Start date/time at the beginning of the hour, choose your Local time zone, and Run frequency. Run frequency will be often the SnapMirror replication will be updated.

   It is recommended to set the log schedule to update every hour to ensure the recovery to the last hourly update.

Restore and Recover Database

Scheduling the Log Replication Playbook

Configure and launch the job template.

1. Copy the previously created job template.
   a. Navigate to Resources → Templates.
   b. Find the ONTAP/CVO Setup Template, and on the far right click on Copy Template
   c. Click Edit Template on the copied template, and change the name to Restore and Recovery Playbook.
   d. Keep the same inventory, project, credentials for the template.
   e. Select the ora_recovery.yml as the playbook to be executed.
   f. The variables will remain the same, but the CVO cluster IP will need to be set in the variable dst_cluster_ip.
   g. Click Save.
This playbook will not be ran until you are ready to restore your database at the remote site.

Recovering Oracle Database

1. On-premises production Oracle databases data volumes are protected via NetApp SnapMirror replication to either a redundant ONTAP cluster in secondary data center or Cloud Volume ONTAP in public cloud. In a fully configured disaster recovery environment, recovery compute instances in secondary data center or public cloud are standby and ready to recover the production database in the case of a disaster. The standby compute instances are kept in sync with on-prem instances by running parallel updates on OS kernel patch or upgrade in a lockstep.

2. In this solution demonstrated, Oracle binary volume is replicated to target and mounted at target instance to bring up Oracle software stack. This approach to recover Oracle has advantage over a fresh installation of Oracle at last minute when a disaster occurred. It guarantees Oracle installation is fully in sync with current on-prem production software installation and patch levels etc. However, this may or may not have additional software licensing implication for the replicated Oracle binary volume at recovery site depending on how the software licensing is structured with Oracle. User is recommended to check with its software licensing personnel to assess the potential Oracle licensing requirement before deciding to use the same approach.

3. The standby Oracle host at the destination is configured with the Oracle prerequisite configurations.

4. The SnapMirrors are broken and the volumes are made writable and mounted to the standby Oracle host.

5. The Oracle recovery module performs following tasks to recovery and startup Oracle at recovery site after all DB volumes are mounted at standby compute instance.
   a. Sync the control file: We deployed duplicate Oracle control files on different database volume to protect critical database control file. One is on the data volume and another is on log volume. Since data and log volumes are replicated at different frequency, they will be out of sync at the time of recovery.
   b. Relink Oracle binary: Since the Oracle binary is relocated to a new host, it needs a relink.
   c. Recover Oracle database: The recovery mechanism retrieves last System Change Number in last available archived log in Oracle log volume from control file and recovers Oracle database to recoup all business transactions that was able to be replicated to DR site at the time of failure. The database is then started up in a new incarnation to carry on user connections and business transaction at recovery site.

Before running the Recovering playbook make sure you have the following:
Make sure it copy over the /etc/oratab and /etc/oraInst.loc from the source Oracle host to the destination host.

TR-4794: Oracle databases on NetApp EF-Series

Mitch Blackburn, Ebin Kadavy, NetApp

TR-4794 is intended to help storage administrators and database administrators successfully deploy Oracle on NetApp EF-Series storage.
Microsoft SQL Server

TR-4951: Backup and Recovery for Microsoft SQL Server on AWS FSx for ONTAP

Author(s): Niyaz Mohammed, Carine Ngwekwe - NetApp Solutions Engineering

This document covers the steps necessary to perform backup and recovery for Microsoft SQL Server on AWS FSx for ONTAP with SnapCenter. This includes the following information:

- NetApp SnapCenter configuration
- SnapCenter backup operations
- Backup operation for an FCI database
- Backup operation for multiple databases
- Restore and recovery

SnapCenter Configuration

The following steps must be performed for SnapCenter configuration and the protection of Microsoft SQL Server resources. Each of the following steps is detailed in the following sections.

1. Configure sysadmin credentials for the SQL Server backup and restore user.
2. Configure storage settings. Provide Amazon Web Services (AWS) management credential to access the Amazon FSx for NetApp ONTAP storage virtual machines (SVMs) from SnapCenter.
3. Add a SQL Server host to SnapCenter. Deploy and install the required SnapCenter Plug-ins.
4. Configure policies. Define the backup operation type, retention, and optional Snapshot backup replication.
5. Configure and protect the Microsoft SQL Server database.

SnapCenter newly installed user interface

Configure credentials for SQL Server backup and restore the user with sysadmin rights.
NetApp recommends using role-based access control (RBAC) to delegate data protection and management capabilities to individual users across the SnapCenter and window hosts. The user must have access to the SQL Server hosting the database. For multiple hosts, the username and password must be the same across the various hosts. Furthermore, to enable SnapCenter to deploy the required plug-in on SQL Server hosts, you must register the domain information for SnapCenter to validate your credentials and hosts.

Expand the following sections to see the detailed instructions on how to complete each step.
Add the credentials

Go to **Settings**, select **Credentials**, and click (+).

The new user must have administrator rights on the SQL Server host.
Configure storage

To configure storage in SnapCenter, complete the following steps:

1. In the SnapCenter UI, select **Storage Systems**. There are two storage types, **ONTAP SVM** and **ONTAP Cluster**. By default, the storage type is **ONTAP SVM**.

2. Click (+) to add the storage system information.

3. Provide the **FSx for ONTAP management** endpoint.

4. The SVM is now configured in SnapCenter.
To add a SQL Server host, complete the following steps:

1. From the Host tab, click (+) to add the Microsoft SQL Server host.

2. Provide the fully qualified domain name (FQDN) or IP address of the remote host.
   - The credentials are populated by default.

3. Select the option for Microsoft Windows and Microsoft SQL Server and then submit.

The SQL Server packages are installed.
1. After the installation is complete, go to the **Resource** tab to verify whether all FSx for ONTAP iSCSI volumes are present.
Configure log directory

To configure a host log directory, complete the following steps:

1. Click the check box. A new tab opens.

2. Click the **configure log directory** link.

3. Select the drive for the host log directory and the FCI instance log directory. Click **Save**. Repeat the same process for the second node in the cluster. Close the window.
The host is now in a running state.

1. From the **Resources** tab, we have all the servers and databases.
Configure a backup policy

A backup policy is a set of rules that govern how to manage, schedule, and retain backup. It helps with the backup type and frequency based on your company’s SLA.

Expand the following sections to see the detailed instructions on how to complete each step.
Configure back-up operation for an FCI database

To configure a backup policy for an FCI database, complete the following steps:

1. Go to **Settings** and select **Policies** on the top left. Then click **New**.

2. Enter the policy name and a description. Click **Next**.

3. Select **Full backup** as the backup type.
4. Select the schedule frequency (this is based on the company SLA). Click **Next**.

5. Configure the retention settings for the backup.
6. Configure the replication options.
7. Specify a run script to run before and after a backup job is run (if any).
8. Run verification based on the backup schedule.
9. The **Summary** page provides details of the backup policy. Any errors can be corrected here.
Configure and protect MSSQL Server database

1. Set up the starting date and expiration date of the backup policy.

2. Define the schedule for the backup. To do that, click (+) to configure a schedule. Enter the **Start date** and **Expires on** date. Set the time based on the company’s SLA.

3. Configure the verification server. From the drop-down menu, select the server.
4. Confirm the configured schedule by clicking the plus sign and confirm.

5. Provide information for email notification. Click Next.

The summary of the backup policy for SQL Server database is now configured.
SnapCenter backup operations

To create on-demand SQL Server backups, complete the following steps:

1. From the **Resource** view, select the resource and select **Backup now**.

2. In the **Backup** dialog box, click **Backup**.
3. A confirmation screen is displayed. Click **Yes** to confirm.

**Monitor backup job**

1. From the **Monitor** tab, click the job and select **Details** on the right to view the jobs.
When the backup is completed, a new entry is shown in the Topology view.

**Backup operation for multiple databases**

To configure a backup policy for multiple SQL Server databases, create resource group policies by completing the following steps:

1. In the Resources tab from the View menu, change to a resource group using the drop-down menu.
2. Click (+) for a new resource group.

3. Provide a name and tag. Click Next.
4. Add resources to the resource group:
   - **Host.** Select the server from the drop-down menu hosting the database.
   - **Resource type.** From the drop-down menu, select **Database**.
   - **SQL Server instance.** Select the server.

The **option Auto Selects All the Resources from the Same Storage Volume** is selected by default. Clear the option and select only the databases you need to add to the resource group, Click the arrow to add and click **Next**.
5. On the policies, click (+).

6. Enter the resource group policy name.
7. Select **Full backup** and the schedule frequency depending on your company’s SLA.

8. Configure the retention settings.
9. Configure the replication options.
10. Configure the scripts to run before performing a backup. Click **Next**.
11. Confirm the verification for the following backup schedules.
12. On the **Summary** page, verify the information, and click **Finish**.
Configure and protect multiple SQL Server databases

1. Click the (+) sign to configure the start date and the expire-on date.

2. Set the time.
Add schedules for policy FULLBACKUP_CHAMPIONSHIPDB

Hourly

Start date: 11/11/2022 05:30 pm
Expires on: 12/11/2022 05:27 pm
Repeat every: 8 hours 0 mins

The schedules are triggered on the SnapCenter Server time zone.
3. From the **Verification** tab, select the server, configure the schedule, and click **Next**.

4. Configure notifications to send an email.
The policy is now configured for backing up multiple SQL Server databases.

**Trigger on-demand backup for multiple SQL Server databases**

1. From the **Resource** tab, select view. From the drop-down menu, select **Resource Group**.
2. Select the resource group name.

3. Click **Backup now** in the upper right.

4. A new window opens. Click the **Verify after backup** checkbox and then click backup.
5. A confirmation message is displayed. Click Yes.

Monitor multiple-database backup jobs

From the left navigation bar, click Monitor, select the backup job, and click Details to view job progress.
Click the **Resource** tab to see the time it takes for the backup to be completed.

**Transaction log backup for multiple database backup**

SnapCenter supports the full, bulked logged, and simple recovery models. The simple recovery mode does not support transactional log backup.

To perform a transaction log backup, complete the following steps:

1. From the **Resources** tab, change the view menu from **Database** to **Resource group**.
2. Select the resource group backup policy created.

3. Select **Modify Resource Group** in the upper right.

4. The **Name** section defaults to the backup policy name and tag. Click **Next**.

The **Resources** tab highlights the bases to which the transaction backup policy is to be configured.
5. Enter the policy name.

6. Select the SQL Server backup options.
7. Select log backup.
8. Set the schedule frequency based on your company’s RTO. Click Next.
9. Configure the log backup retention settings. Click **Next**.
10. (Optional) Configure the replication options.
11. (Optional) Configure any scripts to run before performing a backup job.

12. (Optional) Configure backup verification.
13. On the **Summary** page, click **Finish**.

**Configure and protect multiple MSSQL Server databases**

1. Click the newly created transaction log backup policy.
2. Set the **Start date** and **Expires on** date.

3. Enter the frequency of the log backup policy depending on the SLA, RTP, and RPO. Click OK.

4. You can see both policies. Click **Next**.
5. Configure the verification server.

6. Configure email notification.
7. On the **Summary** page, click **Finish**.

**Triggering an on-demand transaction log backup for multiple SQL Server databases**

To trigger an on-demand backup of the transactional log for multiple SQL server databases, complete the following steps:

1. On the newly created policy page, select **Backup now** at the upper right of the page.
2. From the pop-up on the Policy tab, select the drop-down menu, select the backup policy, and configure the transaction log backup.

3. Click Backup. A new window is displayed.

4. Click Yes to confirm the backup policy.
Monitoring

Move to the Monitoring tab and monitor the progress of the backup job.

Restore and recovery

See the following prerequisites necessary for restoring a SQL Server database in SnapCenter.

• The target instance must be online and running before a restore job completes.
• SnapCenter operations that are scheduled to run against the SQL Server database must be disabled,
including any jobs scheduled on remote management or remote verification servers.

- If you are restoring custom log directory backups to an alternate host, the SnapCenter server and the plugin host must have the same SnapCenter version installed.
- You can restore the system database to an alternate host.
- SnapCenter can restore a database in a Windows cluster without taking the SQL Server cluster group offline.

**Restoring deleted tables on a SQL Server database to a point in time**

To restore a SQL Server database to a point in time, complete the following steps:

1. The following screenshot shows the initial state of the SQL Server database before the deleted tables.

   ![Screenshot of SQL Server database before deletion](image)

   The screenshot shows that 20 rows were deleted from the table.
2. Log into SnapCenter Server. From the **Resources** tab, select the database.

3. Select the most recent backup.

4. On the right, select **Restore**.
5. A new window is displayed. Select the **Restore** option.

6. Restore the database to the same host where the backup was created. Click **Next**.

7. For the **Recovery type**, select **All log backups**. Click **Next**.
Pre-restore options:

1. Select the option **Overwrite the database with same name during restore**. Click Next.
Post-restore options:

1. Select the option **Operational, but unavailable for restoring additional transaction logs**. Click **Next**.

2. Provide the email settings. Click **Next**.
3. On the **Summary** page, click **Finish**.

**Monitoring the restore progress**

1. From the **Monitoring** tab, click the restore job details to view the progress of the restore job.
2. Restore the job details.

3. Return to SQL Server host > database > table are present.
Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

• TR-4714: Best Practices Guide for Microsoft SQL Server using NetApp SnapCenter

• Requirements for restoring a database
  https://docs.netapp.com/us-en/snapcenter-45/protect-scsql/concept_requirements_for_restoring_a_database.html

• Understanding cloned database lifecycles

TR-4923: SQL Server on AWS EC2 using Amazon FSx for NetApp ONTAP

Authors: Pat Sinthusan and Niyaz Mohamed, NetApp

Introduction

Many companies that would like to migrate applications from on-premises to the cloud find that the effort is hindered by the differences in capabilities offered by on-premises storage systems and cloud storage services. That gap has made migrating enterprise applications such as Microsoft SQL Server much more problematic. In particular, gaps in the services needed to run an enterprise application such as robust snapshots, storage efficiency capabilities, high availability, reliability, and consistent performance have forced customers to make design tradeoffs or forgo application migration. With FSx for NetApp ONTAP, customers no longer need to compromise. FSx for NetApp ONTAP is a native (1st party) AWS service sold, supported, billed, and fully managed by AWS. It uses the power of NetApp ONTAP to provide the same enterprise grade storage and data management capabilities NetApp has provided on-premises for three decades in AWS as a managed service.
With SQL Server on EC2 instances, database administrators can access and customize their database environment and the underlying operating system. A SQL Server on EC2 instance in combination with AWS FSx ONTAP to store the database files, enables high performance, data management, and a simple and easy migration path using block-level replication. Therefore, you can run your complex database on AWS VPC with an easy lift-and-shift approach, fewer clicks, and no schema conversions.

Benefits of using Amazon FSx for NetApp ONTAP with SQL Server

Amazon FSx for NetApp ONTAP is the ideal file storage for SQL Server deployments in AWS. Benefits include the following:

- Consistent high performance and throughput with low latency
- Intelligent caching with NVMe cache to improve performance
- Flexible sizing so that you can increase or shrink capacity, throughput, and IOPs on the fly
- Efficient on-premises-to-AWS block replication
- The use of iSCSI, a well-known protocol for the database environment
- Storage efficiency features like thin provisioning and zero-footprint clones
- Backup time reduction from hours to mins, thereby reducing the RTO
- Granular backup and recovery of SQL databases with the intuitive NetApp SnapCenter UI
- The ability to perform multiple test migrations before actual migration
- Shorter downtime during migration and overcoming migration challenges with file-level or I/O-level copy
- Reducing MTTR by finding the root cause after a major release or patch update

Deploying SQL Server databases on FSx ONTAP with the iSCSI protocol, as is commonly used on-premises, provides an ideal database storage environment with superior performance, storage efficiency, and data-management capabilities. Using multiple iSCSI sessions, assuming a 5% working set size, fitting a Flash Cache delivers over 100K IOPs with the FSx ONTAP service. This configuration provides complete control over performance for the most demanding applications. SQL Server running on smaller EC2 instances connected to FSx for ONTAP can perform the same as SQL Server running on a much larger EC2 instance, because only network bandwidth limits are applied against FSx for ONTAP. Reducing the size of instances also reduces the compute cost, which provides a TCO-optimised deployment. The combination of SQL using iSCSI, SMB3.0 with multichannel, continuous availability shares on FSx for ONTAP provides great advantages for SQL workloads.

Before you begin

The combination of Amazon FSx for NetApp ONTAP and SQL Server on EC2 instance enables the creation of enterprise-level database storage designs that can meet today's most demanding application requirements. To optimize both technologies, it is vital to understand SQL Server I/O patterns and characteristics. A well-designed storage layout for a SQL Server database supports the performance of SQL Server and the management of the SQL Server infrastructure. A good storage layout also allows the initial deployment to be successful and the environment to grow smoothly over time as your business grows.

Prerequisites

Before you complete the steps in this document, you should have the following prerequisites:

- An AWS account
- Appropriate IAM roles to provision EC2 and FSx for ONTAP
• A Windows Active Directory domain on EC2
• All SQL Server nodes must be able to communicate with each other
• Make sure DNS resolution works and host names can be resolved. If not, use host file entry.
• General knowledge of SQL Server installation

Also, please refer to the NetApp Best Practices for SQL Server environments to ensure the best storage configuration.
Best practice configurations for SQL Server environments on EC2

With FSx ONTAP, procuring storage is the easiest task and can be performed by updating the file system. This simple process enables dynamic cost and performance optimization as needed, it helps to balance the SQL workload, and it is also a great enabler for thin provisioning. FSx ONTAP thin provisioning is designed to present more logical storage to EC2 instances running SQL Server than what is provisioned in the file system. Instead of allocating space upfront, storage space is dynamically allocated to each volume or LUN as data is written. In most configurations, free space is also released back when data in the volume or LUN is deleted (and is not being held by any Snapshot copies). The following table provides configuration settings for dynamically allocating storage.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume guarantee</td>
<td>None (set by default)</td>
</tr>
<tr>
<td>LUN reservation</td>
<td>Enabled</td>
</tr>
<tr>
<td>fractional_reserve</td>
<td>0% (set by default)</td>
</tr>
<tr>
<td>snap_reserve</td>
<td>0%</td>
</tr>
<tr>
<td>Autodelete</td>
<td>volume / oldest_first</td>
</tr>
<tr>
<td>Autosize</td>
<td>On</td>
</tr>
<tr>
<td>try_first</td>
<td>Autogrow</td>
</tr>
<tr>
<td>Volume tiering policy</td>
<td>Snapshot only</td>
</tr>
<tr>
<td>Snapshot policy</td>
<td>None</td>
</tr>
</tbody>
</table>

With this configuration, the total size of the volumes can be greater than the actual storage available in the file system. If the LUNs or Snapshot copies require more space than is available in the volume, the volumes automatically grow, taking more space from the containing file system. Autogrow allows FSx ONTAP to automatically increase the size of the volume up to a maximum size that you predetermine. There must be space available in the containing file system to support the automatic growth of the volume. Therefore, with autogrow enabled, you should monitor the free space in the containing filesystem and update the file system when needed.

Along with this, set the space-allocation option on LUN to enabled so that FSx ONTAP notifies the EC2 host when the volume has run out of space and the LUN in the volume cannot accept writes. Also, this option enables FSx for ONTAP to reclaim space automatically when the SQL Server on EC2 host deletes data. The space-allocation option is set to disabled by default.

If a space-reserved LUN is created in a none-guaranteed volume, then the LUN behaves the same as a non-space-reserved LUN. This is because a none-guaranteed volume has no space to allocate to the LUN; the volume itself can only allocate space as it is written to due to its none guarantee.

With this configuration, FSx ONTAP administrators can generally size the volume so that they must manage and monitor the used space in the LUN on the host side and in the file system.

NetApp recommends using a separate file system for SQL server workloads. If the file system is used for multiple applications, monitor the space usage of both the file system and volumes within the file system to make sure that volumes are not competing for available space.
Snapshot copies used to create FlexClone volumes are not deleted by the autodelete option.

Overcommitment of storage must be carefully considered and managed for a mission-critical application such as SQL server for which even a minimal outage cannot be tolerated. In such a case, it is best to monitor storage consumption trends to determine how much, if any, overcommitment is acceptable.

**Best Practices**

1. For optimal storage performance, provision file-system capacity to 1.35x times the size of total database usage.
2. Appropriate monitoring accompanied by an effective action plan is required when using thin provisioning to avoid application downtime.
3. Make sure to set Cloudwatch and other monitoring tool alerts so that people are contacted with enough time to react as storage is filled.

**Configure Storage for SQL Server and deploy Snapcenter for Backup, Restore and clone operations**

In order to perform SQL server operations with SnapCenter, you must first create volumes and LUNs for SQL server.
Create volumes and LUNs for SQL Server

To create volumes and LUNs for SQL Server, complete the following steps:

1. Open the Amazon FSx console at https://console.aws.amazon.com/fsx/
2. Create an Amazon FSx for the NetApp ONTAP file system using the Standard Create option under Creation Method. This allows you to define FSxadmin and vsadmin credentials.

**Creation method**

- **Quick create**
  - Use recommended best-practice configurations.
  - Most configuration options can be changed after the file system is created.

- **Standard create**
  - You set all of the configuration options, including specifying performance, networking, security, backups, and maintenance.

1. Specify the password for fsxadmin.

   **File system administrative password**
   - Password for this file system’s “fsxadmin” user, which you can use to access the ONTAP CLI or REST API.
   - Don’t specify a password
   - Specify a password

   Password

   Confirm password

1. Specify the password for SVMs.

   **SVM administrative password**
   - Password for this SVM’s “vsadmin” user, which you can use to access the ONTAP CLI or REST API.
   - Don’t specify a password
   - Specify a password

   Password

   Confirm password

1. Create volumes by following the step listed in Creating a volume on FSx for NetApp ONTAP.

**Best practices**
- Disable storage Snapshot copy schedules and retention policies. Instead, use NetApp SnapCenter to coordinate Snapshot copies of the SQL Server data and log volumes.
- Configure databases on individual LUNs on separate volumes to leverage fast and granular restore functionality.
- Place user data files (.mdf) on separate volumes because they are random read/write workloads. It is common to create transaction log backups more frequently than database backups. For this reason, place transaction log files (.ldf) on a separate volume from the data files so that independent backup schedules can be created for each. This separation also isolates the sequential write I/O of the log files from the random read/write I/O of data files and significantly improves SQL Server performance.
- Tempdb is a system database used by Microsoft SQL Server as a temporary workspace, especially for I/O intensive DBCC CHECKDB operations. Therefore, place this database on a dedicated volume. In large environments in which volume count is a challenge, you can consolidate tempdb into fewer volumes and store it in the same volume as other system databases after careful planning. Data protection for tempdb is not a high priority because this database is recreated every time Microsoft SQL Server is restarted.

2. Use the following SSH command to create volumes:

```
vol create -vserver svm001 -volume vol_awssqlprod01_data -aggregate aggr1 -size 800GB -state online -tiering-policy snapshot-only -percent -snapshot-space 0 -autosize-mode grow -snapshot-policy none -security -style ntfs
volume modify -vserver svm001 -volume vol_awssqlprod01_data -fractional -reserve 0
volume modify -vserver svm001 -volume vol_awssqlprod01_data -space-mgmt -try-first vol_grow
volume snapshot autodelete modify -vserver svm001 -volume vol_awssqlprod01_data -delete-order oldest_first
```

1. Start the iSCSI service with PowerShell using elevated privileges in Windows Servers.

```
Start-service -Name msiscsi
Set-Service -Name msiscsi -StartupType Automatic
```

1. Install Multipath-IO with PowerShell using elevated privileges in Windows Servers.

```
Install-WindowsFeature -name Multipath-IO -Restart
```

1. Find the Windows initiator Name with PowerShell using elevated privileges in Windows Servers.

```
Get-InitiatorPort | select NodeAddress
```
1. Connect to Storage virtual machines (SVM) using putty and create an iGroup.

```bash
igroup create -igroup igrp_ws2019sql1 -protocol iscsi -ostype windows
```

1. Use the following SSH command to create LUNs:

```bash
lun create -path /vol/vol_awssqlprod01_data/lun_awssqlprod01_data -size
700GB -ostype windows_2008 -space-allocation enabled
lun create -path /vol/vol_awssqlprod01_log/lun_awssqlprod01_log -size 100GB -ostype
windows_2008 -space-allocation enabled
```

```
svmsql:~> lun create -path /vol/vol_awssqlprod01_data/lun_awssqlprod01_data -size 700GB -ostype windows_2008
Created a LUN of size 700g (751619276800)
svmsql:~> lun create -path /vol/vol_awssqlprod01_log/lun_awssqlprod01_log -size 100GB -ostype windows_2008
Created a LUN of size 100g (107374182400)
```

1. To achieve I/O alignment with the OS partitioning scheme, use windows_2008 as the recommended LUN type. Refer here for additional information.

2. Use the following SSH command to the map igroup to the LUNs that you just created.

```bash
lun show
lun map -path /vol/vol_awssqlprod01_data/lun_awssqlprod01_data -igroup
igrp_awssqlprod01lun map -path
/vol/vol_awssqlprod01_log/lun_awssqlprod01_log -igroup
igrp_awssqlprod01
```
1. For a shared disk that uses the Windows Failover Cluster, run an SSH command to map the same LUN to the igroup that belong to all servers that participate in the Windows Failover Cluster.

2. Connect Windows Server to an SVM with an iSCSI target. Find the target IP address from AWS Portal.

svmsql (svm-09e98ab33a31b724a)

<table>
<thead>
<tr>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM ID</td>
</tr>
<tr>
<td>SVM name</td>
</tr>
<tr>
<td>UUID</td>
</tr>
<tr>
<td>File system ID</td>
</tr>
<tr>
<td>Resource ARN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Endpoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management DNS name</td>
</tr>
<tr>
<td>NFS DNS name</td>
</tr>
<tr>
<td>iSCSI DNS name</td>
</tr>
</tbody>
</table>

1. From Server Manager and the Tools menu, select the iSCSI Initiator. Select the Discovery tab and then select Discover Portal. Supply the iSCSI IP address from previous step and select Advanced. From Local Adapter, select Microsoft iSCSI Initiator. From Initiator IP, select the IP of the server. Then
1. Repeat step 12 for the second iSCSI IP from the SVM.

2. Select the **Targets** tab, select **Connect**, and select **Enable multi-path**.

3. For best performance, add more sessions; NetApp recommends creating five iSCSI sessions. Select **Properties** ➔ **Add session** ➔ **Advanced** and repeat step 12.
TargetPortals = ('10.2.1.167', '10.2.2.12')
foreach ($TargetPortal in $TargetPortals) {New-IscsiTargetPortal
-TargetPortalAddress $TargetPortal}

Best practices

- Configure five iSCSI sessions per target interface for optimal performance.
- Configure a round-robin policy for the best overall iSCSI performance.
- Make sure that the allocation unit size is set to 64K for partitions when formatting the LUNs

  1. Run the following PowerShell command to make sure that the iSCSI session is persisted.

\$targets = Get-IscsiTarget
deforeach ($target in $targets)
{
  Connect-IscsiTarget -IsMultipathEnabled $true -NodeAddress $target.NodeAddress -IsPersistent $true
}

1. Initialize disks with the following PowerShell command.
```powershell
$disks = Get-Disk | where PartitionStyle -eq raw
foreach ($disk in $disks) {Initialize-Disk $disk.Number}
```

You can automate volume and LUN creation using the PowerShell script from Appendix B. LUNs can also be created using SnapCenter.

Once the volumes and LUNs are defined, you need to set up SnapCenter to be able to perform the database operations.

**SnapCenter overview**

NetApp SnapCenter is next-generation data protection software for tier-1 enterprise applications. SnapCenter, with its single-pane-of-glass management interface, automates and simplifies the manual, complex, and time-consuming processes associated with the backup, recovery, and cloning of multiple databases and other application workloads. SnapCenter leverages NetApp technologies, including NetApp Snapshots, NetApp SnapMirror, SnapRestore, and NetApp FlexClone. This integration allows IT organizations to scale their storage infrastructure, meet increasingly stringent SLA commitments, and improve the productivity of administrators across the enterprise.
SnapCenter Server requirements

The following table lists the minimum requirements for installing the SnapCenter Server and plug-in on Microsoft Windows Server.

<table>
<thead>
<tr>
<th>Components</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum CPU count</td>
<td>Four cores/vCPUs</td>
</tr>
<tr>
<td>Memory</td>
<td>Minimum: 8GB</td>
</tr>
<tr>
<td></td>
<td>Recommended: 32GB</td>
</tr>
<tr>
<td>Storage space</td>
<td>Minimum space for installation: 10GB</td>
</tr>
<tr>
<td></td>
<td>Minimum space for repository: 10GB</td>
</tr>
<tr>
<td>Supported operating system</td>
<td>• Windows Server 2012</td>
</tr>
<tr>
<td></td>
<td>• Windows Server 2012 R2</td>
</tr>
<tr>
<td></td>
<td>• Windows Server 2016</td>
</tr>
<tr>
<td></td>
<td>• Windows Server 2019</td>
</tr>
<tr>
<td>Software packages</td>
<td>• .NET 4.5.2 or later</td>
</tr>
<tr>
<td></td>
<td>• Windows Management Framework (WMF) 4.0 or later</td>
</tr>
<tr>
<td></td>
<td>• PowerShell 4.0 or later</td>
</tr>
</tbody>
</table>

For detailed information, refer to space and sizing requirements.

For version compatibility, see the NetApp Interoperability Matrix Tool.
The following figure depicts some considerations for creating the Microsoft SQL Server database storage layout when backing up with SnapCenter.

**Best practices**

1. Place databases with I/O-intensive queries or with large database size (say 500GB or more) on a separate volume for faster recovery. This volume should also be backed up by separate jobs.
2. Consolidate small-to-medium size databases that are less critical or have fewer I/O requirements to a single volume. Backing up a large number of databases residing in the same volume leads to fewer Snapshot copies that need to be maintained. It is also a best practice to consolidate Microsoft SQL Server instances to use the same volumes to control the number of backup Snapshot copies taken.
3. Create separate LUNs to store full text-related files and file-streaming related files.
4. Assign separate LUNs per host to store Microsoft SQL Server log backups.
5. System databases that store database server metadata configuration and job details are not updated frequently. Place system databases/tempdb in separate drives or LUNs. Do not place system databases in the same volume as the user databases. User databases have a different backup policy, and the frequency of user database backup is not same for system databases.
6. For Microsoft SQL Server Availability Group setup, place the data and log files for replicas in an identical folder structure on all nodes.

In addition to the performance benefit of segregating the user database layout into different volumes, the database also significantly affects the time required to back up and restore. Having separate volumes for data and log files significantly improves the restore time as compared to a volume hosting multiple user data files. Similarly, user databases with a high I/O intensive application are prone to an increase in the backup time. A more detailed explanation about backup and restore practices is provided later in this document.
Starting with SQL Server 2012 (11.x), system databases (Master, Model, MSDB, and TempDB), and Database Engine user databases can be installed with an SMB file server as a storage option. This applies to both stand-alone SQL Server and SQL Server failover cluster installations. This enables you to use FSx for ONTAP with all its performance and data management capabilities, including volume capacity, performance scalability, and data protection features, which SQL Server can take advantage of. Shares used by the application servers must be configured with the continuously available property set and the volume should be created with NTFS security style. NetApp Snapcenter cannot be used with databases placed on SMB shares from FSx for ONTAP.

For SQL Server databases that do not use SnapCenter to perform backups, Microsoft recommends placing the data and log files on separate drives. For applications that simultaneously update and request data, the log file is write intensive, and the data file (depending on your application) is read/write intensive. For data retrieval, the log file is not needed. Therefore, requests for data can be satisfied from the data file placed on its own drive.

When you create a new database, Microsoft recommends specifying separate drives for the data and logs. To move files after the database is created, the database must be taken offline. For more Microsoft recommendations, see Place Data and Log Files on Separate Drives.
Installation and setup for SnapCenter

Follow the **Install the SnapCenter Server** and **Installing SnapCenter Plug-in for Microsoft SQL Server** to install and setup SnapCenter.

After Installing SnapCenter, complete the following steps to set it up.

1. To set up credentials, select **Settings > New** and then enter the credential information.

![Credential settings](image)

1. Add the storage system by selecting **Storage Systems > New** and provide the appropriate FSx for ONTAP storage information.

![Storage system settings](image)

1. Add hosts by selecting **Hosts > Add**, and then provide the host information. SnapCenter automatically installs the Windows and SQL Server plug-in. This process might take some time.
After all Plug-ins are installed, you must configure the log directory. This is the location where the transaction log backup resides. You can configure the log directory by selecting the host and then select configure the log directory.

SnapCenter uses a host log directory to store transaction log backup data. This is at the host and instance level. Each SQL Server host used by SnapCenter must have a host log directory configured to perform log backups. SnapCenter has a database repository, so metadata related to backup, restore, or cloning operations is stored in a central database repository.

The size of the host log directory is calculated as follows:

Size of host log directory = system database size + (maximum DB LDF size × daily log change rate % × (Snapshot copy retention) ÷ (1 – LUN overhead space %)

The host log directory sizing formula assumes the following:

- A system database backup that does not include the tempdb database
- A 10% LUN overhead space

Place the host log directory on a dedicated volume or LUN. The amount of data in the host log directory depends on the size of the backups and the number of days that backups are retained.
If the LUNs have already been provisioned, you can select the mount point to represent the host log directory.

Now you are ready to perform backup, restore and clone operations for SQL Server.
Backup database with SnapCenter

After placing the database and log files on the FSx ONTAP LUNs, SnapCenter can be used to back up the databases. The following processes are used to create a full backup.

**Best Practices**

- In SnapCenter terms, RPO can be identified as the backup frequency, for example, how frequently you want to schedule the backup so that you can reduce the loss of data to up to few minutes. SnapCenter allows you to schedule backups as frequently as every five minutes. However, there might be a few instances in which a backup might not complete within five minutes during peak transaction times or when the rate of change of data is more in the given time. A best practice is to schedule frequent transaction log backups instead of full backups.

- There are numerous approaches to handle the RPO and RTO. One alternative to this backup approach is to have separate backup policies for data and logs with different intervals. For example, from SnapCenter, schedule log backups in 15-minute intervals and data backups in 6-hour intervals.

- Use a resource group for a backup configuration for Snapshot optimization and the number of jobs to be managed.

1. Select **Resources**, and then select **Microsoft SQL Server** on the drop-down menu on the top left. Select **Refresh Resources**.

2. Select the database to be backed up, then select **Next** and (*) to add the policy if one has not been created. Follow the **New SQL Server Backup Policy** to create a new policy.

3. Select the verification server if necessary. This server is the server that SnapCenter runs DBCC CHECKDB after a full backup has been created. Click **Next** for notification, and then select **Summary** to review. After reviewing, click **Finish**.
4. Click **Back up Now** to test the backup. In the pop-up windows, select **Backup**.

5. Select **Monitor** to verify that the backup has been completed.

**Best Practices**
• Backup the transaction log backup from SnapCenter so that during the restoration process, SnapCenter can read all the backup files and restore in sequence automatically.

• If third party products are used for backup, select Copy backup in SnapCenter to avoid log sequence issues, and test the restore functionality before rolling into production.

## Restore database with SnapCenter

One of the major benefits of using FSx ONTAP with SQL Server on EC2 is its ability to perform fast and granular restore at each database level.

Complete the following steps to restore an individual database to a specific point in time or up to the minute with SnapCenter.

1. Select Resources and then select the database that you would like to restore.

2. Select the backup name that the database needs to be restored from and then select restore.

3. Follow the Restore pop-up windows to restore the database.

4. Select Monitor to verify that the restore process is successful.
Considerations for an instance with a large number of small-to-large size databases

SnapCenter can back up a large number of sizeable databases in an instance or group of instances within a resource group. The size of a database is not the major factor in backup time. The duration of a backup can vary depending on number of LUNs per volume, the load on Microsoft SQL Server, the total number of databases per instance, and, specifically, the I/O bandwidth and usage. While configuring the policy to back up databases from an instance or resource group, NetApp recommends that you restrict the maximum database backed up per Snapshot copy to 100 per host. Make sure the total number of Snapshot copies does not exceed the 1,023-copy limit.

NetApp also recommends that you limit the backup jobs running in parallel by grouping the number of databases instead of creating multiple jobs for each database or instance. For optimal performance of the backup duration, reduce the number of backup jobs to a number that can back up around 100 or fewer databases at a time.

As previously mentioned, I/O usage is an important factor in the backup process. The backup process must wait to quiesce until all the I/O operations on a database are complete. Databases with highly intensive I/O operations should be deferred to another backup time or should be isolated from other backup jobs to avoid affecting other resources within the same resource group that are to be backed up.

For an environment that has six Microsoft SQL Server hosts hosting 200 databases per instance, assuming four LUNs per host and one LUN per volume created, set the full backup policy with the maximum databases backed up per Snapshot copy to 100. Two hundred databases on each instance are laid out as 200 data files distributed equally on two LUNs, and 200 log files are distributed equally on two LUNs, which is 100 files per LUN per volume.

Schedule three backup jobs by creating three resource groups, each grouping two instances that include a total of 400 databases.

Running all three backup jobs in parallel backs up 1,200 databases simultaneously. Depending on the load on the server and I/O usage, the start and end time on each instance can vary. In this instance, a total of 24 Snapshot copies are created.

In addition to the full backup, NetApp recommends that you configure a transaction log backup for critical databases. Make sure that the database property is set to full recovery model.

Best practices

1. Do not include the tempdb database in a backup because the data it contains is temporary. Place tempdb on a LUN or an SMB share that is in a storage system volume in which Snapshot copies will not be created.

2. A Microsoft SQL Server instance with a high I/O intensive application should be isolated in a different backup job to reduce the overall backup time for other resources.

3. Limit the set of databases to be simultaneously backed up to approximately 100 and stagger the remaining set of database backups to avoid a simultaneous process.

4. Use the Microsoft SQL Server instance name in the resource group instead of multiple databases because whenever new databases are created in Microsoft SQL Server instance, SnapCenter automatically considers a new database for backup.

5. If you change the database configuration, such as changing the database recovery model to the full recovery model, perform a backup immediately to allow up-to-the-minute restore operations.

6. SnapCenter cannot restore transaction log backups created outside of SnapCenter.

7. When cloning FlexVol volumes, make sure that you have sufficient space for the clone metadata.
8. When restoring databases, make sure that sufficient space is available on the volume.
9. Create a separate policy to manage and back up system databases at least once a week.
Cloning databases with SnapCenter

To restore a database onto another location on a dev or test environment or to create a copy for business analysis purposes, the NetApp best practice is to leverage the cloning methodology to create a copy of the database on the same instance or an alternate instance.

The cloning of databases that are 500GB on an iSCSI disk hosted on a FSx for ONTAP environment typically takes less than five minutes. After cloning is complete, the user can then perform all the required read/write operation on the cloned database. Most of the time is consumed for disk scanning (diskpart). The NetApp cloning procedure typically take less than 2 minutes regardless of the size of the databases.

The cloning of a database can be performed with the dual method: you can create a clone from the latest backup or you can use clone life-cycle management through which the latest copy can be made available on the secondary instance.

SnapCenter allows you to mount the clone copy on the required disk to maintain the format of the folder structure on the secondary instance and continue to schedule backup jobs.

Clone databases to the new database name in the same instance

The following steps can be used to clone databases to the new database name in the same SQL server instance running on EC2:

1. Select Resources and then the database that need to be cloned.
2. Select the backup name that you would like to clone and select Clone.
3. Follow the clone instructions from the backup windows to finish the clone process.
4. Select Monitor to make sure that cloning is completed.
Clone databases into the new SQL Server instance running on EC2

The following steps are used to clone databases to the new SQL Server instance running on EC2:

1. Create a new SQL Server on EC2 in the same VPC.
2. Enable the iSCSI protocol and MPIO, and then setup the iSCSI connection to FSx for ONTAP by following steps 3 and 4 in the section “Create volumes and LUNs for SQL Server.”
3. Add a new SQL Server on EC2 into SnapCenter by follow step 3 in the section “Installing and setup for SnapCenter.”
4. Select Resource > View Instance, and then select Refresh Resource.
5. Select Resources, and then the database that you would like to clone.
6. Select the backup name that you would like to clone, and then select Clone.

1. Follow the Clone from Backup instructions by providing the new SQL Server instance on EC2 and instance name to finish the clone process.
2. Select Monitor to make sure that cloning is completed.

To learn more about this process, watch the following video:

Clone databases into the new SQL Server instance running on EC2

Appendices

Appendix A: YAML file for use in Cloud Formation Template

The following .yaml file can be used with the Cloud Formation Template in AWS Console.

- https://github.com/NetApp/fsxn-iscsisetup-cft

To automate iSCSI LUN creation and NetApp SnapCenter installation with PowerShell, clone the repo from this GitHub link.
Appendix B: Powershell scripts for provisioning volumes and LUNs

The following script is used to provision volumes and LUNs and also to set up iSCSI based on the instruction provided above. There are two PowerShell scripts:

* _EnableMPIO.ps1

```
Function Install_MPIO_ssh {
    $hostname = $env:COMPUTERNAME
    $hostname = $hostname.Replace('-','_')
    #Add schedule action for the next step
    $path = Get-Location
    $path = $path.Path + '\2_CreateDisks.ps1'
    $arg = '-NoProfile -WindowStyle Hidden -File ' +$path
    $schAction = New-ScheduledTaskAction -Execute "Powershell.exe" -Argument $arg
    $schTrigger = New-ScheduledTaskTrigger -AtStartup
    $schPrincipal = New-ScheduledTaskPrincipal -UserId "NT AUTHORITY\SYSTEM" -LogonType ServiceAccount -RunLevel Highest
    $return = Register-ScheduledTask -Action $schAction -Trigger $schTrigger -TaskName "Create Vols and LUNs" -Description "Scheduled Task to run configuration Script At Startup" -Principal $schPrincipal
    #Install -Module Posh-SSH
    Write-host 'Enable MPIO and SSH for PowerShell' -ForegroundColor Yellow
    $return = Find-PackageProvider -Name 'Nuget' -ForceBootstrap -IncludeDependencies
    $return = Find-Module PoSH-SSH | Install-Module -Force
    #Install Multipath-IO with PowerShell using elevated privileges in Windows Servers
    Write-host 'Enable MPIO' -ForegroundColor Yellow
    $return = Install-WindowsFeature -name Multipath-IO -Restart
}
Install_MPIO_ssh
Remove-Item -Path $MyInvocation.MyCommand.Source
```

* _CreateDisks.ps1

```
....
#Enable MPIO and Start iSCSI Service
Function PrepISCSI {
    $return = Enable-MSDSMAutomaticClaim -BusType iSCSI
    #Start iSCSI service with PowerShell using elevated privileges in Windows Servers
```
$return = Start-service -Name msiscsi
$return = Set-Service -Name msiscsi -StartupType Automatic

Function Create_igroup_vols_luns ($fsxN){
    $hostname = $env:COMPUTERNAME
    $hostname = $hostname.Replace('-','_')
    $volsluns = @()
    for ($i = 1;$i -lt 10;$i++){
        if ($i -eq 9){
            $volsluns +=(@{volname=('v_'+$hostname+'_log');volsize=$fsxN.logvolsize;lunname=('l_'+$hostname+'_log');lunsize=$fsxN.loglunsize})
        } else {
            $volsluns +=(@{volname=('v_'+$hostname+'_data'+[string]$i);volsize=$fsxN.datavolsizen;lunname=('l_'+$hostname+'_data'+[string]$i);lunsize=$fsxN.datalunsize})
        }
    }
    $secStringPassword = ConvertTo-SecureString $fsxN.password -AsPlainText -Force
    $igroup = 'igrp_'+$hostname
    #Connect to FSx N filesystem
    $session = New-SSHSession -ComputerName $fsxN.svmip -Credential $credObject -AcceptKey:$true
    #Create igroup
    Write-host 'Creating igroup' -ForegroundColor Yellow
    #Find Windows initiator Name with PowerShell using elevated privileges in Windows Servers
    $initport = Get-InitiatorPort | select -ExpandProperty NodeAddress
    $sshcmd = 'igroup create -igroup ' + $igroup + ' -protocol iscsi -ostype windows -initiator ' + $initport
    $ret = Invoke-SSHCommand -Command $sshcmd -SSHSession $session
    #Create vols
    Write-host 'Creating Volumes' -ForegroundColor Yellow
    foreach ($vollun in $volsluns){
        $sshcmd = 'vol create ' + $vollun.volname + ' -aggregate aggr1 -size ' + $vollun.volsize #+ ' -vserver ' + $vserver
        $return = Invoke-SSHCommand -Command $sshcmd -SSHSession $session
    }
    #Create LUNs and mapped LUN to igroup
    Write-host 'Creating LUNs and map to igroup' -ForegroundColor Yellow
foreach ($vollun in $volsluns)
{
    $sshcmd = "lun create -path /vol/" + $vollun.volname + "/" + $vollun.lunname + " -size " + $vollun.lunsize + " -ostype Windows_2008 "="#vserver " +$vserver
    $return = Invoke-SSHCommand -Command $sshcmd -SSHSession $session
    #map all luns to igroup
    $sshcmd = "lun map -path /vol/" + $vollun.volname + "/" + $vollun.lunname + " -igroup " + $igroup
    $return = Invoke-SSHCommand -Command $sshcmd -SSHSession $session
}
}

Function Connect_iSCSI_to_SVM ($TargetPortals)
{
    Write-host 'Online, Initialize and format disks' -ForegroundColor Yellow
    #Connect Windows Server to svm with iSCSI target.
    foreach ($TargetPortal in $TargetPortals) {
        New-IscsiTargetPortal -TargetPortalAddress $TargetPortal
        for ($i = 1; $i -lt 5; $i++){
            $return = Connect-IscsiTarget -IsMultipathEnabled $true -IsPersistent $true -NodeAddress (Get-iscsiTarget | select -ExpandProperty NodeAddress)
        }
    }
}

Function Create_Partition_Format_Disks{
    #Create Partion and format disk
    $disks = Get-Disk | where PartitionStyle -eq raw
    foreach ($disk in $disks) {
        $return = Initialize-Disk $disk.Number
        $partition = New-Partition -DiskNumber $disk.Number -AssignDriveLetter -UseMaximumSize | Format-Volume -FileSystem NTFS -AllocationUnitSize 65536 -Confirm:$false -Force
        #$return = Format-Volume -DriveLetter $partition.DriveLetter -FileSystem NTFS -AllocationUnitSize 65536
    }
}

Function UnregisterTask {
    Unregister-ScheduledTask -TaskName "Create Vols and LUNs" -Confirm:$false
}

Start-Sleep -s 30
$fsxN = @{svmip = "198.19.255.153"; login = 'vsadmin'; password = 'net@pp11'; datavolsize = '10GB'; datalunsize = '8GB'}
Run the file `EnableMPIO.ps1` first and the second script executes automatically after the server has been rebooted. These PowerShell scripts can be removed after they have been executed due to credential access to the SVM.

Where to find additional information

- Amazon FSx for NetApp ONTAP
  https://docs.aws.amazon.com/fsx/latest/ONTAPGuide/what-is-fsx-ontap.html
- Getting Started with FSx for NetApp ONTAP
  https://docs.aws.amazon.com/fsx/latest/ONTAPGuide/getting-started.html
- Overview of the SnapCenter interface
  https://www.youtube.com/watch?v=IVEBF4kV6Ag&t=0s
- Tour through SnapCenter navigation pane options
  https://www.youtube.com/watch?v=_lDKt-koySQ
- Setup SnapCenter 4.0 for SQL Server plug-in
  https://www.youtube.com/watch?v=MopbUFSdHKE
- How to back up and restore databases using SnapCenter with SQL Server plug-in
  https://www.youtube.com/watch?v=K343qPD5_Ys
- How to clone a database using SnapCenter with SQL Server plug-in
  https://www.youtube.com/watch?v=ogEc4DkGv1E

**TR-4897: SQL Server on Azure NetApp Files - Real Deployment View**

Niyaz Mohamed, NetApp

IT organizations face constant change. Gartner reports nearly 75% of all databases will require cloud-based storage by 2022. As a leading relational database management system (RDBMS), Microsoft SQL Server is the go-to choice for Windows platform-designed applications and organizations that rely on SQL Server for everything from enterprise resource planning (ERP) to analytics to content management. SQL Server has
helped to revolutionize the way enterprises manage massive data sets and power their applications to meet the schema and query performance demands.

Most IT organizations follow a cloud-first approach. Customers in a transformation phase evaluate their current IT landscape and then migrate their database workloads to the cloud based on an assessment and discovery exercise. Some factors driving customers toward cloud migration include elasticity/burst, data center exit, data center consolidation, end-of-life scenarios, mergers, acquisitions, and so on. The reason for migration can vary based on each organization and their respective business priorities. When moving to the cloud, choosing the right cloud storage is very important in order to unleash the power of SQL Server database cloud deployment.

**Use case**

Moving the SQL Server estate to Azure and integrating SQL Server with Azure’s vast array of platform-as-a-service (PaaS) features such as Azure Data Factory, Azure IoT Hub, and Azure Machine Learning creates tremendous business value to support digital transformation. Adopting the cloud also enables the respective business unit to focus on productivity and delivering new features and enhancements faster (DevTest use case) than relying on the CAPEX model or traditional private cloud models. This document covers a real-time deployment of SQL Server Always On availability group (AOAG) on Azure NetApp Files leveraging Azure Virtual Machines.

Azure NetApp Files provides enterprise-grade storage with continuously available file shares. Continuously available shares are required by SQL Server production databases on SMB file share to make sure that the node always has access to the database storage, including during disruptive scenarios such as controller upgrades or failures. Continuously available file shares eliminate the need to replicate data between storage nodes. Azure NetApp Files uses SMB 3.0 scale-out, persistent handles, and transparent failover to support nondisruptive operations (NDOs) for planned and unplanned downtime events, including many administrative tasks.

When planning cloud migrations, you should always evaluate the best approach to use. The most common and easiest approach for application migration is rehosting (also known as lift and shift). The example scenario provided in this document uses the rehosting method. SQL Server on Azure virtual machines with Azure NetApp Files allows you to use full versions of SQL Server in the cloud without having to manage on-premises hardware. SQL Server virtual machines (VMs) also simplify licensing costs when you pay as you go and provides elasticity and bursting capabilities for development, test, and estate refresh scenarios.

**Factors to consider**

**VM performance**

Selecting the right VM size is important for optimal performance of a relational database in a public cloud. Microsoft recommends that you continue using the same database performance-tuning options that are applicable to SQL Server in on-premises server environments. Use memory-optimized VM sizes for the best performance of SQL Server workloads. Collect the performance data of existing deployment to identify the RAM and CPU utilization while choosing the right instances. Most deployments choose between the D, E, or M series.

**Notes:**

- For the best performance of SQL Server workloads, use memory-optimized VM sizes.
- NetApp and Microsoft recommend that you identify the storage performance requirements before choosing the instance type with the appropriate memory-to-vCore ratio. This also helps select a lower-instance type with the right network bandwidth to overcome storage throughput limits of the VM.
VM redundancy

To increase redundancy and high availability, SQL Server VMs should either be in the same availability set or different availability zones. When creating Azure VMs, you must choose between configuring availability sets versus availability zones; an Azure VM cannot participate in both.

High availability

For high availability, configuring SQL Server AOAG or Always On Failover Cluster Instance (FCI) is the best option. For AOAG, this involves multiple instances of SQL Server on Azure Virtual Machines in a virtual network. If high availability is required at the database level, consider configuring SQL Server availability groups.

Storage configuration

Microsoft SQL Server can be deployed with an SMB file share as the storage option. Starting with SQL Server 2012, system databases (master, model, msdb, or tempdb), and user databases can be installed with Server Message Block (SMB) file server as a storage option. This applies to both SQL Server stand-alone and SQL Server FCI.

File share storage for SQL Server databases should support continuously available property. This provides uninterrupted access to the file-share data.

Azure NetApp Files provides high performing file storage to meet any demanding workload, and it reduces SQL Server TCO as compared to block storage solutions. With block storage, VMs have imposed limits on I/O and bandwidth for disk operations; network bandwidth limits alone are applied against Azure NetApp Files. In other words, no VM-level I/O limits are applied to Azure NetApp Files. Without these I/O limits, SQL Server running on smaller VMs connected to Azure NetApp Files can perform as well as SQL Server running on much larger VMs. Azure NetApp Files reduce SQL Server deployment costs by reducing compute and software licensing costs. For detailed cost analysis and performance benefits of using Azure NetApp Files for SQL Server deployment, see the Benefits of using Azure NetApp Files for SQL Server deployment.

Benefits

The benefits of using Azure NetApp Files for SQL Server include the following:

- Using Azure NetApp Files allows you to use smaller instances, thus reducing compute cost.
- Azure NetApp Files also reduces software licensing costs, which reduce the overall TCO.
- Volume reshaping and dynamic service level capability optimizes cost by sizing for steady-state workloads and avoiding overprovisioning.

Notes:

- To increase redundancy and high availability, SQL Server VMs should either be in the same availability set or in different availability zones. Consider file path requirements if user-defined data files are required; in which case, select SQL FCI over SQL AOAG.
- The following UNC path is supported: \\ANFSMB-b4ca.anf.test\SQLDB and \\ANFSMB-b4ca.anf.test\SQLDB\. 
- The loopback UNC path is not supported.
- For sizing, use historic data from your on-premises environment. For OLTP workloads, match the target IOPS with performance requirements using workloads at average and peak times along with the disk reads/sec and disk writes/sec performance counters. For data warehouse and reporting workloads, match
Create continuously available shares

Create continuously available shares with the Azure portal or Azure CLI. In the portal, select the Enable Continuous Availability property option. For the Azure CLI, specify the share as a continuously available share by using the `az netappfiles volume create` command with the `smb-continuously-avl` option set to `$True`. To learn more about creating a new, continuous availability-enabled volume, see Creating a Continuously Available Share.

Notes:

- Enable continuous availability for the SMB volume as shown in the following image.
- If a non-administrator domain account is used, make sure the account has the required security privilege assigned.
- Set the appropriate permissions at the share level and proper file-level permissions.
- A continuously available property cannot be enabled on existing SMB volumes. To convert an existing volume to use a continuously available share, use NetApp Snapshot technology. For more information, see Convert existing SMB volumes to use Continuous Availability.

Create a volume

Configure access to your volume.

**Access**

Protocol type

- NFS
- SMB
- Dual-protocol (NFsv3 and SMB)

**Configuration**

Active Directory

Share name

Enable Continuous Availability

Performance

Azure NetApp Files supports three service levels: Standard (16MBps per terabyte), Premium (64MBps per terabyte), and Ultra (128MBps per terabyte). Provisioning the right volume size is important for optimal performance.
performance of the database workload. With Azure NetApp Files, volume performance and the throughput limit are based on a combination of the following factors:

- The service level of the capacity pool to which the volume belongs
- The quota assigned to the volume
- The quality of service (QoS) type (auto or manual) of the capacity pool

For more information, see Service levels for Azure NetApp Files.

Performance validation

As with any deployment, testing the VM and storage is critical. For storage validation, tools such as HammerDB, Apploader, the SQL Server storage benchmark (SB) tool, or any custom script or FIO with the appropriate read/write mix should be used. Keep in mind however that most SQL Server workloads, even busy OLTP workloads, are closer to 80%–90% read and 10%–20% write.

To showcase performance, a quick test was performed against a volume using premium service levels. In this test, the volume size was increased from 100GB to 2TB on the fly without any disruption to application access and zero data migration.
Here is another example of real time performance testing with HammerDB performed for the deployment covered in this paper. For this testing, we used a small instance with eight vCPUs, a 500GB Premium SSD, and a 500GB SMB Azure NetApp Files volume. HammerDB was configured with 80 warehouses and eight users.

The following chart shows that Azure NetApp Files was able to deliver 2.6x the number of transactions per minute at 4x lower latency when using a comparable sized volume (500GB).

An additional test was performed by resizing to a larger instance with 32x vCPUs and a 16TB Azure NetApp Files volume. There was a significant increase in transactions per minute with consistent 1ms latency. HammerDB was configured with 80 warehouses and 64 users for this test.
Cost optimization

Azure NetApp Files allows nondisruptive, transparent volume resizing and the ability to change the service levels with zero downtime and no effect on applications. This is a unique capability allowing dynamic cost management that avoids the need to perform database sizing with peak metrics. Rather, you can use steady state workloads, which avoids upfront costs. The volume reshaping and dynamic service-level change allows you to adjust the bandwidth and service level of Azure NetApp Files volumes on demand almost instantaneously without pausing I/O, while retaining data access.

Azure PaaS offerings such as LogicApp or Functions can be used to easily resize the volume based on a specific webhook or alert rule trigger to meet the workload demands while dynamically handling the cost.

For example, consider a database that needs 250MBps for steady state operation; however, it also requires a peak throughput of 400MBps. In this case, the deployment should be performed with a 4TB volume within the Premium service level to meet the steady-state performance requirements. To handle the peak workload, increase the volume size using Azure functions to 7TB for that specific period, and then downsize the volume to make the deployment cost effective. This configuration avoids overprovisioning of the storage.

Real-time, high-level reference design

This section covers a real-time deployment of a SQL database estate in an AOAG configuration using an Azure NetApp Files SMB volume.

- Number of nodes: 4
- Number of databases: 21
- Number of availability groups: 4
- Backup retention: 7 days
- Backup archive: 365 days

Deploying FCI with SQL Server on Azure virtual machines with an Azure NetApp Files share provides a cost-efficient model with a single copy of the data. This solution can prevent add-file operation issues if the file path differs from the secondary replica.
The following image shows the databases within AOAG spread across the nodes.

### Data layout

The user database files (.mdf) and user database transaction log files (.ldf) along with tempDB are stored on the same volume. The service level is Ultra.
The configuration consists of four nodes and four AGs. All 21 databases (part of Dynamic AX, SharePoint, RDS connection broker, and indexing services) are stored on the Azure NetApp Files volumes. The databases are balanced between the AOAG nodes to use the resources on the nodes effectively. Four D32 v3 instances are added in the WSFC, which participates in the AOAG configuration. These four nodes are provisioned in the Azure virtual network and are not migrated from on-premises.

Notes:

- If the logs require more performance and throughput depending on the nature of the application and the queries executed, the database files can be placed on the Premium service level, and the logs can be stored at the Ultra service level.
- If the tempdb files have been placed on Azure NetApp Files, then the Azure NetApp Files volume should be separated from the user database files. Here is an example distribution of the database files in AOAG.

Notes:

- To retain the benefits of Snapshot copy-based data protection, NetApp recommends not combining data and log data into the same volume.
- An add-file operation performed on the primary replica might fail on the secondary databases if the file path of a secondary database differs from the path of the corresponding primary database. This can happen if the share path is different on primary and secondary nodes (due to different computer accounts). This failure could cause the secondary databases to be suspended. If the growth or performance pattern cannot be predicted and the plan is to add files later, a SQL Server failover cluster with Azure NetApp Files is an acceptable solution. For most deployments, Azure NetApp Files meets the performance requirements.

Migration

There are several ways to migrate an on-premises SQL Server user database to SQL Server in an Azure virtual machine. The migration can be either online or offline. The options chosen depend on the SQL Server version, business requirements, and the SLAs defined within the organization. To minimize downtime during the database migration process, NetApp recommends using either the AlwaysOn option or the transactional replication option. If it is not possible to use these methods, you can migrate the database manually.

The simplest and most thoroughly tested approach for moving databases across machines is backup and restore. Typically, you can start with a database backup followed by a copy of the database backup into Azure. You can then restore the database. For the best data transfer performance, migrate the database files into the Azure VM using a compressed backup file. The high-level design referenced in this document uses the backup approach to Azure file storage with Azure file sync and then restore to Azure NetApp files.

Azure Migrate can be used to discover, assess, and migrate SQL Server workloads.

To perform a migration, complete the following high-level steps:

1. Based on your requirements, set up connectivity.
2. Perform a full database backup to an on-premises file-share location.
3. Copy the backup files to an Azure file share with Azure file sync.
4. Provision the VM with the desired version of SQL Server.
5. Copy the backup files to the VM by using the `copy` command from a command prompt.
6. Restore the full databases to SQL Server on Azure virtual machines.
To restore 21 databases, it took approximately nine hours. This approach is specific to this scenario. However, other migration techniques listed below can be used based on your situation and requirements.

Other migration options to move data from an on-premises SQL Server to Azure NetApp Files include the following:

- Detach the data and log files, copy them to Azure Blob storage, and then attach them to SQL Server in the Azure VM with an ANF file share mounted from the URL.
- If you are using Always On availability group deployment on-premises, use the Add Azure Replica Wizard to create a replica in Azure and then perform failover.
- Use SQL Server transactional replication to configure the Azure SQL Server instance as a subscriber, disable replication, and point users to the Azure database instance.
- Ship the hard drive using the Windows Import/Export Service.

**Backup and recovery**

Backup and recovery are an important aspect of any SQL Server deployment. It is mandatory to have the appropriate safety net to quickly recover from various data failure and loss scenarios in conjunction with high availability solutions such as AOAG. SQL Server Database Quiesce Tool, Azure Backup (streaming), or any third-party backup tool such as Commvault can be used to perform an application-consistent backup of the databases.

Azure NetApp Files Snapshot technology allows you to easily create a point-in-time (PiT) copy of the user databases without affecting performance or network utilization. This technology also allows you to restore a Snapshot copy to a new volume or quickly revert the affected volume to the state it was in when that Snapshot copy was created by using the revert volume function. The Azure NetApp Files snapshot process is very quick and efficient, which allows for multiple daily backups, unlike the streaming backup offered by Azure backup. With multiple Snapshot copies possible in a given day, the RPO and RTO times can be significantly reduced. To add application consistency so that data is intact and properly flushed to the disk before the Snapshot copy is taken, use the SQL Server database quiesce tool ([SCSQLAPI tool](https://docs.microsoft.com/en-us/sql-server/availability/database-quiesce?view=sql-server-2017); access to this link requires NetApp SSO login credentials). This tool can be executed from within PowerShell, which quiesces the SQL Server database and in turn can take the application-consistent storage Snapshot copy for backups.

*Notes:*

- The SCSQLAPI tool only supports the 2016 and 2017 versions of SQL Server.
- The SCSQLAPI tool only works with one database at a time.
- Isolate the files from each database by placing them onto a separate Azure NetApp Files volume.

Because of SCSQL API’s vast limitations, Azure Backup was used for data protection in order to meet the SLA requirements. It offers a stream-based backup of SQL Server running in Azure Virtual Machines and Azure NetApp Files. Azure Backup allows a 15-minute RPO with frequent log backups and PiT recovery up to one second.

**Monitoring**

Azure NetApp Files is integrated with Azure Monitor for the time series data and provides metrics on allocated storage, actual storage usage, volume IOPS, throughput, disk read bytes/sec, disk write bytes/sec, disk reads/sec and disk writes/sec, and associated latency. This data can be used to identify bottlenecks with alerting and to perform health checks to verify that your SQL Server deployment is running in an optimal configuration.
In this HLD, ScienceLogic is used to monitor Azure NetApp Files by exposing the metrics using the appropriate service principal. The following image is an example of the Azure NetApp Files Metric option.

DevTest using thick clones

With Azure NetApp Files, you can create instantaneous copies of databases to test functionality that should be implemented by using the current database structure and content during the application development cycles, to use the data extraction and manipulation tools when populating data warehouses, or to even recover data that was mistakenly deleted or changed. This process does not involve copying data from Azure Blob containers, which makes it very efficient. After the volume is restored, it can be used for read/write operations, which significantly reduces validation and time to market. This needs to be used in conjunction with SCSQLAPI for application consistency. This approach provides yet another continuous cost optimization technique along with Azure NetApp Files leveraging the Restore to New volume option.

Notes:

• The volume created from the Snapshot copy using the Restore New Volume option consumes capacity from the capacity pool.
• You can delete the cloned volumes by using REST or Azure CLI to avoid additional costs (in case the capacity pool must be increased).

Hybrid storage options

Although NetApp recommends using the same storage for all the nodes in SQL Server availability groups, there are scenarios in which multiple storage options can be used. This scenario is possible for Azure NetApp Files in which a node in AOAG is connected with an Azure NetApp Files SMB file share and the second node is connected with an Azure Premium disk. In these instances, make sure that the Azure NetApp Files SMB share is holding the primary copy of the user databases and the Premium disk is used as the secondary copy.

Notes:

• In such deployments, to avoid any failover issues, make sure that continuous availability is enabled on the SMB volume. With no continuously available attribute, the database can fail if there is any background maintenance at the storage layer.
• Keep the primary copy of the database on the Azure NetApp Files SMB file share.

Business continuity

Disaster recovery is generally an afterthought in any deployment. However, disaster recovery must be addressed during the initial design and deployment phase to avoid any impact to your business. With Azure
NetApp Files, the cross-region replication (CRR) functionality can be used to replicate the volume data at the block level to the paired region to handle any unexpected regional outage. The CRR-enabled destination volume can be used for read operations, which makes it an ideal candidate for disaster recovery simulations. In addition, the CRR destination can be assigned with the lowest service level (for instance, Standard) to reduce the overall TCO. In the event of a failover, replication can be broken, which makes the respective volume read/write capable. Also, the service level of the volume can be changed by using the dynamic service level functionality to significantly reduce disaster recovery cost. This is another unique feature of Azure NetApp Files with block replication within Azure.

**Long-term Snapshot copy archive**

Many organizations must perform long-term retention of snapshot data from database files as a mandatory compliance requirement. Although this process is not used in this HLD, it can be easily accomplished by using a simple batch script using AzCopy to copy the snapshot directory to the Azure Blob container. The batch script can be triggered based on a specific schedule by using scheduled tasks. The process is straightforward—it includes the following steps:

1. Download the AzCopy V10 executable file. There is nothing to install because it is an `exe` file.
2. Authorize AzCopy by using a SAS token at the container level with the appropriate permissions.
3. After AzCopy is authorized, the data transfer begins.

**Notes:**

- In batch files, make sure to escape the `%` characters that appear in SAS tokens. This can be done by adding an additional `%` character next to existing `%` characters in the SAS token string.
- The Secure Transfer Required setting of a storage account determines whether the connection to a storage account is secured with Transport Layer Security (TLS). This setting is enabled by default. The following batch script example recursively copies data from the Snapshot copy directory to a designated Blob container:

```
SET source="Z:\~snapshot"
echo %source%
SET dest="https://testanfacct.blob.core.windows.net/azcoptst?sp=racwdl&st=2020-10-21T18:41:35Z&se=2021-10-22T18:41:00Z&sv=2019-12-12&sr=c&sig=ZxRUJwFlXgHS8As7HzXJOaDXXVJ7PxxIX3ACpx56XY%3D"
echo %dest%
```

The following example cmd is executed in PowerShell:

```
-recurse
```
Notes:

- A similar backup feature for long-term retention will soon be available in Azure NetApp Files.
- The batch script can be used in any scenario that requires data to copied to Blob container of any region.

Cost optimization

With volume reshaping and dynamic service level change, which is completely transparent to the database, Azure NetApp Files allows continuous cost optimizations in Azure. This capability is used in this HLD extensively to avoid overprovisioning of additional storage to handle workload spikes.

Resizing the volume can be easily accomplished by creating an Azure function in conjunction with the Azure alert logs.

Conclusion

Whether you are targeting an all-cloud or hybrid cloud with stretch databases, Azure NetApp Files provides excellent options to deploy and manage the database workloads while reducing your TCO by making data requirements seamless to the application layer.

This document covers recommendations for planning, designing, optimizing, and scaling Microsoft SQL Server deployments with Azure NetApp Files, which can vary greatly between implementations. The right solution depends on both the technical details of the implementation and the business requirements driving the project.

Takeaways

The key points of this document include:

- You can now use Azure NetApp Files to host the database and file share witness for SQL Server cluster.
• You can boost the application response times and deliver 99.9999% availability to provide access to SQL Server data when and where it is needed.
• You can simplify the overall complexity of the SQL Server deployment and ongoing management, such as raid striping, with simple and instant resizing.
• You can rely on intelligent operations features to help you deploy SQL Server databases in minutes and speed development cycles.
• If Azure Cloud is the destination, Azure NetApp Files is the right storage solution for optimized deployment.

Where to find additional information

To learn more about the information described in this document, refer to the following website links:

• Solution architectures using Azure NetApp Files

• Benefits of using Azure NetApp Files for SQL Server deployment

• SQL Server on Azure Deployment Guide Using Azure NetApp Files

• Fault tolerance, high availability, and resilience with Azure NetApp Files

TR-4467: SAP with Microsoft SQL Server on Windows - Best practices using NetApp Clustered Data ONTAP and SnapCenter

Marco Schoen, NetApp

TR-4467 provides customers and partners with best practices for deploying clustered NetApp Data ONTAP in support of SAP Business Suite solutions running in a Microsoft SQL Server on Windows environment.

TR-4467: SAP with Microsoft SQL Server on Windows - Best practices using NetApp Clustered Data ONTAP and SnapCenter

Modernizing your Microsoft SQL Server environment

Optimize operations and unleash the power of your data - on the premises or in the cloud.

TR-4590: Best practice guide for Microsoft SQL Server with ONTAP

Manohar Kulkarni and Pat Sinthusan, NetApp

This document describes best practices and offers insight into design considerations for deploying SQL Server on NetApp storage systems running NetApp ONTAP® software, with the goal of achieving effective and efficient storage deployment and end-to-end data protection and retention planning.
Open Source Databases

**TR-4956: Automated PostgreSQL High Availability Deployment and Disaster Recovery in AWS FSx/EC2**

Allen Cao, Niyaz Mohamed, NetApp

**Purpose**

PostgreSQL is a widely used open-source database that is ranked number four among the top ten most popular database engines by DB-Engines. On one hand, PostgreSQL derives its popularity from its license-free, open-source model while still possessing sophisticated features. On the other hand, because it is open sourced, there is shortage of detailed guidance on production-grade database deployment in the area of high availability and disaster recovery (HA/DR), particularly in the public cloud. In general, it can be difficult to set up a typical PostgreSQL HA/DR system with hot and warm standby, streaming replication, and so on. Testing the HA/DR environment by promoting the standby site and then switching back to the primary can be disruptive to production. There are well documented performance issues on the primary when read workloads are deployed on streaming hot standby.

In this documentation, we demonstrate how you can do away with an application-level PostgreSQL streaming HA/DR solution and build a PostgreSQL HA/DR solution based on AWS FSx ONTAP storage and EC2 compute instances using storage-level replication. The solution creates a simpler and comparable system and delivers equivalent results when compared with traditional PostgreSQL application-level streaming replication for HA/DR.

This solution is built on proven and mature NetApp SnapMirror storage-level replication technology that is available in AWS-native FSx ONTAP cloud storage for PostgreSQL HA/DR. It is simple to implement with an automation toolkit provided by the NetApp Solutions team. It provides similar functionality while eliminating the complexity and performance drag on the primary site with the application-level streaming-based HA/DR solution. The solution can be easily deployed and tested without affecting the active primary site.

This solution addresses the following use cases:

- Production grade HA/DR deployment for PostgreSQL in the public AWS cloud
- Testing and validating a PostgreSQL workload in the public AWS cloud
- Testing and validating a PostgreSQL HA/DR strategy based on NetApp SnapMirror replication technology

**Audience**

This solution is intended for the following people:
• The DBA who is interested in deploying PostgreSQL with HA/DR in the public AWS cloud.
• The database solution architect who is interested in testing PostgreSQL workloads in the public AWS cloud.
• The storage administrator who is interested in deploying and managing PostgreSQL instances deployed to AWS FSx storage.
• The application owner who is interested in standing up a PostgreSQL environment in AWS FSx/EC2.

Solution test and validation environment

The testing and validation of this solution was performed in an AWS FSx and EC2 environment that might not match the final deployment environment. For more information, see the section [Key Factors for Deployment Consideration].

Architecture

PostgreSQL hybrid cloud solution architecture with AWS/FSx

Hardware and software components

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSx ONTAP storage</td>
<td>on-prem Centos VM/4vCPU/8G</td>
</tr>
<tr>
<td>Current version</td>
<td>A VM to host Ansible automation controller either on-premise or in the cloud</td>
</tr>
<tr>
<td>Two FSx HA pairs in the same VPC and availability zone as primary and standby HA clusters</td>
<td>Two EC2 T2 xlarge as primary and standby compute instances</td>
</tr>
<tr>
<td>t2.xlarge/4vCPU/16G</td>
<td></td>
</tr>
<tr>
<td>EC2 instance for compute</td>
<td></td>
</tr>
<tr>
<td>Ansible controller</td>
<td></td>
</tr>
<tr>
<td>RedHat Linux</td>
<td>RHEL-8.6.0_HVM-20220503-x86_64-2-Hourly2-GP2</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>Centos Linux</td>
<td>CentOS Linux release 8.2.2004 (Core)</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Version 14.5</td>
</tr>
<tr>
<td>Ansible</td>
<td>Version 2.10.3</td>
</tr>
</tbody>
</table>

Key factors for deployment consideration

- **PostgreSQL database backup, restore, and recovery.** A PostgreSQL database supports a number of backup methods, such as a logical backup using pg_dump, a physical online backup with pg_basebackup or a lower-level OS backup command, and storage-level-consistent snapshots. This solution uses NetApp consistency-group snapshots for PostgreSQL database data and WAL volumes backup, restore, and recovery at the standby site. The NetApp consistency-group volume snapshots sequence I/O as it is written to storage and protect the integrity of database data files.

- **EC2 compute instances.** In these tests and validations, we used the AWS EC2 t2.xlarge instance type for the PostgreSQL database compute instance. NetApp recommends using an M5 type EC2 instance as the compute instance for PostgreSQL in deployment because it is optimized for database workloads. The standby compute instance should always be deployed in the same zone as the passive (standby) file system deployed for the FSx HA cluster.

- **FSx storage HA clusters single- or multi-zone deployment.** In these tests and validations, we deployed an FSx HA cluster in a single AWS availability zone. For production deployment, NetApp recommends deploying an FSx HA pair in two different availability zones. A disaster-recovery standby HA pair for business continuity can be set up in a different region if a specific distance is required between the primary and standby. An FSx HA cluster is always provisioned in a HA pair that is sync mirrored in a pair of active-passive file systems to provide storage-level redundancy.

- **PostgreSQL data and log placement.** Typical PostgreSQL deployments share the same root directory or volumes for data and log files. In our tests and validations, we have separated PostgreSQL data and logs into two separate volumes for performance. A soft link is used in the data directory to point to the log directory or volume that hosts PostgreSQL WAL logs and archived WAL logs.

- **PostgreSQL service startup delay timer.** This solution uses NFS mounted volumes to store the PostgreSQL database file and WAL log files. During a database host reboot, PostgreSQL service might try to start while the volume is not mounted. This results in database service startup failure. A 10 to 15 seconds timer delay is needed for the PostgreSQL database to start up correctly.

- **RPO/RTO for business continuity.** FSx data replication from primary to standby for DR is based on ASYNC, which means that the RPO depends on the frequency of Snapshot backups and SnapMirror replication. A higher frequency of Snapshot copy and SnapMirror replication reduces the RPO. Therefore, there is a balance between potential data loss in the event of a disaster and incremental storage cost. We have determined that Snapshot copy and SnapMirror replication can be implemented in as low as 5 minute intervals for RPO, and PostgreSQL can generally be recovered at the DR standby site in under a minute for the RTO.

- **Database backup.** After a PostgreSQL database is implemented or migrated into AWS FSx storage from an on-premises data center, the data is auto-sync mirrored in the FSx HA pair for protection. Data is further protected with a replicated standby site in case of a disaster. For longer-term backup retention or data protection, NetApp recommends using the built-in PostgreSQL pg_basebackup utility to run a full
Solution Deployment

The deployment of this solution can be completed automatically using the NetApp Ansible-based automation toolkit by following the detailed instructions outlined below.

1. Read the instructions in the automation toolkit README.md na_postgresql_aws_deploy_hadr.
2. Watch the following video walk through.

Automated PostgreSQL Deployment and Protection

1. Configure the required parameters files (hosts, host_vars/host_name.yml, fsx_vars.yml) by entering user-specific parameters into the template in the relevant sections. Then use the copy button to copy files to the Ansible controller host.

Prerequisites for automated deployment

Deployment requires the following prerequisites.

1. An AWS account has been set up, and the necessary VPC and network segments have been created within your AWS account.
2. From the AWS EC2 console, you must deploy two EC2 Linux instances, one as the primary PostgreSQL DB server at the primary and one at the standby DR site. For compute redundancy at the primary and standby DR sites, deploy two additional EC2 Linux instances as standby PostgreSQL DB servers. See the architecture diagram in the previous section for more details about the environment setup. Also review the User Guide for Linux instances for more information.
3. From the AWS EC2 console, deploy two FSx ONTAP storage HA clusters to host the PostgreSQL database volumes. If you are not familiar with the deployment of FSx storage, see the documentation Creating FSx for ONTAP file systems for step-by-step instructions.
4. Build a Centos Linux VM to host the Ansible controller. The Ansible controller can be located either on-premises or in the AWS cloud. If it is located on-premises, you must have SSH connectivity to the VPC, EC2 Linux instances, and FSx storage clusters.
5. Set up the Ansible controller as described in the section "Set up the Ansible Control Node for CLI deployments on RHEL/CentOS" from the resource Getting Started with NetApp solution automation.
6. Clone a copy of the automation toolkit from the public NetApp GitHub site.

```bash
git clone https://github.com/NetApp-Automation/na_postgresql_aws_deploy_hadr.git
```

1. From the toolkit root directory, execute the prerequisite playbooks to install the required collections and libraries for the Ansible controller.

```bash
ansible-playbook -i hosts requirements.yml
```
ansible-galaxy collection install -r collections/requirements.yml --force --force-with-deps

1. Retrieve the required EC2 FSx instance parameters for the DB host variables file `host_vars/*` and the global variables file `fsx_vars.yml` configuration.

Configure the hosts file

Input the primary FSx ONTAP cluster management IP and EC2 instances hosts names into the hosts file.

```yaml
# Primary FSx cluster management IP address
[fsx_ontap]
172.30.15.33

# Primary PostgreSQL DB server at primary site where database is initialized at deployment time
[postgresql]
psql_01p ansible_ssh_private_key_file=psql_01p.pem

# Primary PostgreSQL DB server at standby site where postgresql service is installed but disabled at deployment
# Standby DB server at primary site, to setup this server comment out other servers in [dr_postgresql]
# Standby DB server at standby site, to setup this server comment out other servers in [dr_postgresql]
[dr_postgresql] --
psql_01s ansible_ssh_private_key_file=psql_01s.pem
#psql_01ps ansible_ssh_private_key_file=psql_01ps.pem
#psql_01ss ansible_ssh_private_key_file=psql_01ss.pem
```

Configure the `host_name.yml` file in the `host_vars` folder
# Add your AWS EC2 instance IP address for the respective PostgreSQL server host
ansible_host: "10.61.180.15"

# "{{groups.postgresql[0]}}" represents first PostgreSQL DB server as defined in PostgreSQL hosts group [postgresql]. For concurrent multiple PostgreSQL DB servers deployment, [0] will be incremented for each additional DB server. For example, "{{groups.postgresql[1]}}" represents DB server 2, "{{groups.postgresql[2]}}" represents DB server 3 ... As a good practice and the default, two volumes are allocated to a PostgreSQL DB server with corresponding /pgdata, /pglogs mount points, which store PostgreSQL data, and PostgreSQL log files respectively. The number and naming of DB volumes allocated to a DB server must match with what is defined in global fsx_vars.yml file by src_db_vols, src_archivelog_vols parameters, which dictates how many volumes are to be created for each DB server. aggr_name is aggr1 by default. Do not change. lif address is the NFS IP address for the SVM where PostgreSQL server is expected to mount its database volumes. Primary site servers from primary SVM and standby servers from standby SVM.

host_datastores_nfs:
  - {vol_name: "{{groups.postgresql[0]}}_pgdata", aggr_name: "aggr1", lif: "172.21.94.200", size: "100"}
  - {vol_name: "{{groups.postgresql[0]}}_pglogs", aggr_name: "aggr1", lif: "172.21.94.200", size: "100"}

# Add swap space to EC2 instance, that is equal to size of RAM up to 16G max. Determine the number of blocks by dividing swap size in MB by 128.
swap_blocks: "128"

# Postgresql user configurable parameters
psql_port: "5432"
buffer_cache: "8192MB"
archive_mode: "on"
max_wal_size: "5GB"
client_address: "172.30.15.0/24"

Configure the global fsx_vars.yml file in the vars folder

########################################################################
######  PostgreSQL HADR global user configuration variables       ######
######  Consolidate all variables from FSx, Linux, and postgresql ######
########################################################################

###########################################
### Ontap env specific config variables ###

1283
# Variables for SnapMirror Peering

# Passphrase for cluster peering authentication
passphrase: "xxxxxxx"

dst_cluster_name: "FsxId0cf8e0bccbl4805e8"

dst_cluster_ip: "172.30.15.90"

dst_inter_ip: "172.30.15.13"

dst_vserver: "dr"

dst_vserver_mgmt_lif: "172.30.15.88"

dst_nfs_lif: "172.30.15.88"

src_cluster_name: "FsxId0cf8e0bccbl4805e8"

src_cluster_ip: "172.30.15.20"

src_inter_ip: "172.30.15.5"

src_vserver: "prod"

src_vserver_mgmt_lif: "172.30.15.115"
PostgreSQL deployment and HA/DR setup

The following tasks deploy the PostgreSQL DB server service and initialize the database at the primary site on the primary EC2 DB server host. A standby primary EC2 DB server host is then set up at the standby site. Finally, DB volume replication is set up from the primary-site FSx cluster to the standby-site FSx cluster for disaster recovery.

1. Create DB volumes on the primary FSx cluster, and set up postgresql on the primary EC2 instance host.
2. Set up the standby DR EC2 instance host.

```
ansible-playbook -i hosts postgresql_standby_setup.yml -u ec2-user --private-key psql_01s.pem -e @vars/fsx_vars.yml
```

3. Set up FSx ONTAP cluster peering and database volume replication.

```
ansible-playbook -i hosts fsx_replication_setup.yml -e @vars/fsx_vars.yml
```

4. Consolidate the previous steps into a single-step PostgreSQL deployment and HA/DR setup.

```
ansible-playbook -i hosts postgresql_hadr_setup.yml -u ec2-user -e @vars/fsx_vars.yml
```

5. For setting up a standby PostgreSQL DB host at either the primary or standby sites, comment out all other servers in the hosts file [dr_postgresql] section and then execute the postgresql_standby_setup.yml playbook with the respective target host (such as psql_01ps or standby EC2 compute instance at primary site). Make sure that a host parameters file such as psql_01ps.yml is configured under the host_vars directory.

```
[dr_postgresql] --
#psql_01s ansible_ssh_private_key_file=psql_01s.pem
psql_01ps ansible_ssh_private_key_file=psql_01ps.pem
#psql_01ss ansible_ssh_private_key_file=psql_01ss.pem
```

```
ansible-playbook -i hosts postgresql_standby_setup.yml -u ec2-user --private-key psql_01ps.pem -e @vars/fsx_vars.yml
```

**PostgreSQL database snapshot backup and replication to standby site**

PostgreSQL database snapshot backup and replication to the standby site can be controlled and executed on the Ansible controller with a user-defined interval. We have validated that the interval can be as low as 5 minutes. Therefore, in the case of failure at the primary site, there is 5 minutes of potential data loss if failure occurs right before the next scheduled snapshot backup.

```
*/15 * * * * /home/admin/na_postgresql_aws_deploy_hadr/data_log_snap.sh
```
Failover to Standby Site for DR

For testing the PostgreSQL HA/DR system as a DR exercise, execute failover and PostgreSQL database recovery on the primary standby EC2 DB instance on standby site by executing following playbook. In an actually DR scenario, execute the same for an actually failover to DR site.

```
ansible-playbook -i hosts postgresql_failover.yml -u ec2-user --private-key psql_01s.pem -e @vars/fsx_vars.yml
```

Resync Replicated DB volumes after Failover Test

Run resync after the failover test to reestablish database-volume SnapMirror replication.

```
ansible-playbook -i hosts postgresql_standby_resync.yml -u ec2-user --private-key psql_01s.pem -e @vars/fsx_vars.yml
```

Failover from primary EC2 DB server to standby EC2 DB server due to EC2 compute instance failure

NetApp recommends running manual failover or using well-established OS cluster-ware that might require a license.

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

- Amazon FSx for NetApp ONTAP

- Amazon EC2
  [https://aws.amazon.com/pm/ec2/?trk=36c6da98-7b20-48fa-8225-4784bcd9843&sc_channel=ps&s_kwcid=ALI442213i467723097970le!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARisAJzsrdqwQrgnh6I7j1jWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcb:G:s&s_kwcid=ALI442213i467723097970le!!g!!aws%20ec2](https://aws.amazon.com/pm/ec2/?trk=36c6da98-7b20-48fa-8225-4784bcd9843&sc_channel=ps&s_kwcid=ALI442213i467723097970le!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARisAJzsrdqwQrgnh6I7j1jWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcb:G:s&s_kwcid=ALI442213i467723097970le!!g!!aws%20ec2)

- NetApp Solution Automation

**TR-4722: MySQL Database on NetApp ONTAP Best Practices**

Anup Bharti, Manohar Kulkarni, Jeffrey Steiner NetApp

MySQL and its variants, including MariaDB and Percona, are widely used for many enterprise applications. These applications range from global social networking sites and massive e-commerce systems to SMB hosting systems containing thousands of database instances. This document describes the configuration requirements and provides guidance on tuning and storage configuration for deploying MySQL on NetApp® ONTAP® data management software. To determine whether the environment, configurations, and versions specified in this report support your environment, consult the Interoperability Matrix Tool (IMT).
SnapCenter for Databases

TR-4988: Oracle Database Backup, Recovery, and Clone on ANF with SnapCenter

Allen Cao, Niyaz Mohamed, NetApp

Purpose

NetApp SnapCenter software is an easy-to-use enterprise platform to securely coordinate and manage data protection across applications, databases, and file systems. It simplifies backup, restore, and clone lifecycle management by offloading these tasks to application owners without sacrificing the ability to oversee and regulate activity on the storage systems. By leveraging storage-based data management, it enables increased performance and availability, as well as reduced testing and development times.

In TR-4987, Simplified, Automated Oracle Deployment on Azure NetApp Files with NFS, we demonstrate automated Oracle deployment on Azure NetApp Files (ANF) in Azure cloud. In this documentation, we showcase Oracle database protection and management on ANF in Azure cloud with a very user-friendly SnapCenter UI tool.

This solution addresses the following use cases:

• Backup and recovery of Oracle database deployed on ANF in Azure cloud with SnapCenter.
• Manage database snapshots and clone copies to accelerate application development and improve data lifecycle management.

Audience

This solution is intended for the following people:

• A DBA who would like to deploy Oracle databases on Azure NetApp Files.
• A database solution architect who would like to test Oracle workloads on Azure NetApp Files.
• A storage administrator who would like to deploy and manage Oracle databases on Azure NetApp Files.
• An application owner who would like to stand up an Oracle database on Azure NetApp Files.

Solution test and validation environment

The testing and validation of this solution were performed in a lab setting that might not match the final deployment environment. See the section [Key Factors for Deployment Consideration] for more information.

Architecture
Hardware and software components

### Hardware

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azure NetApp Files</td>
<td>Current offering in Azure by Microsoft</td>
</tr>
<tr>
<td></td>
<td>A capacity pool with Premium service level</td>
</tr>
<tr>
<td>Azure VM for DB server</td>
<td>Standard_B4ms - 4 vCPUs, 16GiB</td>
</tr>
<tr>
<td></td>
<td>Two Linux virtual machine instances</td>
</tr>
<tr>
<td>Azure VM for SnapCenter</td>
<td>Standard_B4ms - 4 vCPUs, 16GiB</td>
</tr>
<tr>
<td></td>
<td>One Windows virtual machine instance</td>
</tr>
</tbody>
</table>

### Software

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RedHat Linux</td>
<td>RHEL Linux 8.6 (LVM) - x64 Gen2</td>
</tr>
<tr>
<td></td>
<td>Deployed RedHat subscription for testing</td>
</tr>
<tr>
<td>Windows Server</td>
<td>2022 DataCenter; AE Hotpatch - x64 Gen2</td>
</tr>
<tr>
<td></td>
<td>Hosting SnapCenter server</td>
</tr>
<tr>
<td>Oracle Database</td>
<td>Version 19.18</td>
</tr>
<tr>
<td></td>
<td>Patch p34765931_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle OPatch</td>
<td>Version 12.2.0.1.36</td>
</tr>
<tr>
<td></td>
<td>Patch p6880880_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>SnapCenter Server</td>
<td>Version 5.0</td>
</tr>
<tr>
<td></td>
<td>Workgroup deployment</td>
</tr>
<tr>
<td>Open JDK</td>
<td>Version java-11-openjdk</td>
</tr>
<tr>
<td></td>
<td>SnapCenter plugin requirement on DB VMs</td>
</tr>
<tr>
<td>NFS</td>
<td>Version 3.0</td>
</tr>
<tr>
<td></td>
<td>Oracle dNFS enabled</td>
</tr>
<tr>
<td>Ansible</td>
<td>core 2.16.2</td>
</tr>
<tr>
<td></td>
<td>Python 3.6.8</td>
</tr>
</tbody>
</table>
Oracle database configuration in the lab environment

<table>
<thead>
<tr>
<th>Server</th>
<th>Database</th>
<th>DB Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora-01</td>
<td>NTAP1(NTAP1_PDB1,NTAP1_PDB2,NTAP1_PDB3)</td>
<td>/u01, /u02, /u03 NFS mounts on ANF capacity pool</td>
</tr>
<tr>
<td>ora-02</td>
<td>NTAP2(NTAP2_PDB1,NTAP2_PDB2,NTAP2_PDB3)</td>
<td>/u01, /u02, /u03 NFS mounts on ANF capacity pool</td>
</tr>
</tbody>
</table>

Key factors for deployment consideration

- **SnapCenter deployment.** SnapCenter can deploy in a Windows domain or Workgroup environment. For domain-based deployment, the domain user account should be a domain administrator account, or the domain user belongs to the local administrator’s group on the SnapCenter hosting server.

- **Name resolution.** SnapCenter server needs to resolve the name to the IP address for each managed target database server host. Each target database server host must resolve the SnapCenter server name to the IP address. If a DNS server is unavailable, add naming to local host files for resolution.

- **Resource group configuration.** Resource group in SnapCenter is a logical grouping of similar resources that can be backed up together. Thus, it simplifies and reduces the number of backup jobs in a large database environment.

- **Separate full database and archive log backup.** Full database backup includes data volumes and log volumes consistent group snapshots. A frequent full database snapshot incurs higher storage consumption but improves RTO. An alternative is less frequent full database snapshots and more frequent archive logs backup, which consumes less storage and improves RPO but may extend RTO. Consider your RTO and RPO objectives when setting up the backup scheme. There is also a limit (1023) of the number of snapshot backups on a volume.

- **Privileges delegation.** Leverage role based access control that is built-in within SnapCenter UI to delegate privileges to application and database teams if desired.

Solution deployment

The following sections provide step-by-step procedures for SnapCenter deployment, configuration, and Oracle database backup, recovery, and clone on Azure NetApp Files in the Azure cloud.

Prerequisites for deployment
Deployment requires existing Oracle databases running on ANF in Azure. If not, follow the steps below to create two Oracle databases for solution validation. For details of Oracle database deployment on ANF in Azure cloud with automation, referred to TR-4987: Simplified, Automated Oracle Deployment on Azure NetApp Files with NFS

1. An Azure account has been set up, and the necessary VNet and network segments have been created within your Azure account.

2. From the Azure cloud portal, deploy Azure Linux VMs as Oracle DB servers. Create an Azure NetApp Files capacity pool and database volumes for Oracle database. Enable VM SSH private/public key authentication for azureuser to DB servers. See the architecture diagram in the previous section for details about the environment setup. Also referred to Step-by-Step Oracle deployment procedures on Azure VM and Azure NetApp Files for detailed information.

   For Azure VMs deployed with local disk redundancy, ensure that you have allocated at least 128G in the VM root disk to have sufficient space to stage Oracle installation files and add OS swap file. Expand /tmplv and /rootlv OS partition accordingly. Ensure the database volume naming follows the VMname-u01, VMname-u02, and VMname-u03 convention.

```
sudo lvresize -r -L +20G /dev/mapper/rootvg-rootlv
```

```
sudo lvresize -r -L +10G /dev/mapper/rootvg-tmplv
```

3. From the Azure cloud portal, provision a Windows server to run the NetApp SnapCenter UI tool with the latest version. Refer to the following link for details: Install the SnapCenter Server.

4. Provision a Linux VM as the Ansible controller node with the latest version of Ansible and Git installed. Refer to the following link for details: Getting Started with NetApp solution automation in section - Setup the Ansible Control Node for CLI deployments on RHEL / CentOS or Setup the Ansible Control Node for CLI deployments on Ubuntu / Debian.

   The Ansible controller node can locate either on-premises or in Azure cloud as far as it can reach Azure DB VMs via ssh port.

5. Clone a copy of the NetApp Oracle deployment automation toolkit for NFS. Follow instructions in TR-4887 to execute the playbooks.

```
git clone https://bitbucket.ngage.netapp.com/scm/ns-bb/na_oracle_deploy_nfs.git
```

6. Stage following Oracle 19c installation files on Azure DB VM /tmp/archive directory with 777 permission.
installer_archives:
- "LINUX.X64_193000_db_home.zip"
- "p34765931_190000_Linux-x86-64.zip"
- "p6880880_190000_Linux-x86-64.zip"

7. Watch the following video:
   Oracle Database Backup, Recovery, and Clone on ANF with SnapCenter

8. Review the Get Started online menu.

SnapCenter installation and setup
We recommend to go through online SnapCenter Software documentation before proceeding to SnapCenter installation and configuration. Following provides a high level summary of steps for installation and setup of SnapCenter software for Oracle on Azure ANF.

1. From SnapCenter Windows server, download and install latest java JDK from Get Java for desktop applications.

2. From SnapCenter Windows server, download and install latest version (currently 5.0) of SnapCenter installation executable from NetApp support site: NetApp | Support.

3. After SnapCenter server installation, launch browser to login to SnapCenter with Windows local admin user or domain user credential via port 8146.

4. Review Get Started online menu.

5. In Settings-Global Settings, check Hypervisor Settings and click on Update.
6. If needed, adjust Session Timeout for SnapCenter UI to the desired interval.

7. Add additional users to SnapCenter if needed.

8. The Roles tab list the built-in roles that can be assigned to different SnapCenter users. Custom roles also can be created by admin user with desired privileges.
9. From Settings–Credential, create credentials for SnapCenter management targets. In this demo use case, they are linux user for login to Azure VM and ANF credential for capacity pool access.
10. From Storage Systems tab, add Azure NetApp Files with credential created above.
11. From Hosts tab, add Azure DB VMs, which installs SnapCenter plugin for Oracle on Linux.

Add Host

- Host Type: Linux
- Host Name: ora-01
- Credentials: azureuser

Select Plug-ins to Install

- Oracle Database
- SAP HANA
- Unix File Systems

More Options: Port, Install Path, Custom Plug-Ins...

Submit  Cancel
12. Once host plugin is installed on DB server VM, databases on the host are auto discovered and visible in Resources tab. Back to Settings-Policies, create backup policies for full Oracle database online backup and archive logs only backup. Refer to this document Create backup policies for Oracle databases for detailed step by step procedures.
A NetApp snapshot backup creates a point-in-time image of the database volumes that you can use to restore in case of a system failure or data loss. Snapshot backups take very little time, usually less than a minute. The backup image consumes minimal storage space and incurs negligible performance overhead because it records only changes to files since the last snapshot copy was made. Following section demonstrates the implementation of snapshots for Oracle database backup in SnapCenter.

1. Navigating to Resources tab, which lists the databases discovered once SnapCenter plugin installed on database VM. Initially, the Overall Status of database shows as Not protected.

2. Click on View drop-down to change to Resource Group. Click on Add sign on the right to add a Resource Group.

3. Name your resource group, tags, and any custom naming.
4. Add resources to your Resource Group. Grouping of similar resources can simplify database management in a large environment.

5. Select the backup policy and set a schedule by clicking on the '+' sign under Configure Schedules.
Add schedules for policy Oracle full online backup

Hourly

- Start date: 02/06/2024 05:55 pm
- Expires on: 03/06/2024 05:51 pm
- Repeat every: 2 hours 0 mins

The schedules are triggered in the SnapCenter Server time zone.

Cancel OK
6. If backup verification is not configured in policy, leave verification page as is.

7. In order to email a backup report and notification, a SMTP mail server is needed in the environment. Or leave it black if a mail server is not setup.

8. Summary of new resource group.
9. Repeat the above procedures to create a database archive log only backup with corresponding backup policy.

10. Click on a resource group to reveal the resources it includes. Besides the scheduled backup job, an one-off backup can be triggered by clicking on Backup Now.
11. Click on the running job to open a monitoring window, which allows the operator to track the job progress in real-time.
12. A snapshot backup set appears under database topology once a successful backup job finishes. A full database backup set includes a snapshot of the database data volumes and a snapshot of the database log volumes. A log-only backup contains only a snapshot of the database log volumes.
Database recovery
Database recovery via SnapCenter restores a snapshot copy of the database volume image point-in-time. The database is then rolled forward to a desired point by SCN/timestamp or a point as allowed by available archive logs in the backup set. The following section demonstrates the workflow of database recovery with SnapCenter UI.

1. From Resources tab, open the database Primary Backup(s) page. Choose the snapshot of database data volume, then click on Restore button to launch database recovery workflow. Note the SCN number or timestamp in the backup sets if you like to run the recovery by Oracle SCN or timestamp.

![Image of SnapCenter UI showing Primary Backup(s) page]

<table>
<thead>
<tr>
<th>Backup Name</th>
<th>Snapshot Lock Expiration</th>
<th>Count</th>
<th>Type</th>
<th>End Date</th>
<th>Verified</th>
<th>Mounted</th>
<th>RMAN Cataloged</th>
<th>SCN</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora-01-02-06-2024_18_00_06_0562_1</td>
<td></td>
<td>1</td>
<td>Log</td>
<td>02/06/2024 6:00:01 PM</td>
<td>Not Applicable</td>
<td>False</td>
<td>Not Cataloged</td>
<td>3374900</td>
</tr>
<tr>
<td>ora-01-02-06-2024_18_00_06_0562_0</td>
<td></td>
<td>1</td>
<td>Data</td>
<td>02/06/2024 6:00:26 PM</td>
<td>Unverified</td>
<td>False</td>
<td>Not Cataloged</td>
<td>3374903</td>
</tr>
<tr>
<td>ora-01-02-06-2024_17_59_01_1158_1</td>
<td></td>
<td>1</td>
<td>Log</td>
<td>02/06/2024 5:59:18 PM</td>
<td>Not Applicable</td>
<td>False</td>
<td>Not Cataloged</td>
<td>3374762</td>
</tr>
</tbody>
</table>

2. Select Restore Scope. For a container database, SnapCenter is flexible to perform a full container database (All Datafiles), pluggable databases, or tablespace level restore.
3. Select **Recovery Scope**. All logs means to apply all available archive logs in the backup set. Point-in-time recovery by SCN or timestamp are also available.
4. The **PreOps** allows execution of scripts against database before restore/recovery operation.
5. The **PostOps** allows execution of scripts against database after restore/recovery operation.
6. Notification via email if desired.
7. Restore job summary
8. Click on running job to open Job Details window. The job status can also be opened and viewed from the Monitor tab.
Database clone
Database clone via SnapCenter is accomplished by creating a new volume from a snapshot of a volume. The system uses the snapshot information to clone a new volume using the data on the volume when the snapshot was taken. More importantly, it is quick (a few minutes) and efficient compared with other methods to make a cloned copy of the production database to support development or testing. Thus, dramatically improve your database application lifecycle management. The following section demonstrates the workflow of database clone with SnapCenter UI.

1. From Resources tab, open the database Primary Backup(s) page. Choose the snapshot of database data volume, then click on clone button to launch database clone workflow.

2. Name the clone database SID. Optionally, for a container database, clone can be done at PDB level as well.
3. Select the DB server where you want to place your cloned database copy. Keep the default file locations unless you want to name them differently.
4. Identical Oracle software stack as in source database should have been installed and configured on clone DB host. Keep the default credential but change Oracle Home Settings to match with settings on clone DB host.
5. The **PreOps** allows execution of scripts before clone operation. Database parameters can be adjusted to meet a clone DB needs as versus a production database, such as reduced SGA target.
6. The **PostOps** allows execution of scripts against database after clone operation. Clone database recovery can be SCN, timestamp based, or Until cancel (rolling forward database to last archived log in the backup set).
7. Notification via email if desired.
8. Clone job summary.
9. Click on running job to open Job Details window. The job status can also be opened and viewed from the Monitor tab.
10. Cloned database registers with SnapCenter immediately.

11. Validate clone database on DB server host. For a cloned development database, database archive mode should be turned off.
[azureuser@ora-02 ~]$ sudo su

[root@ora-02 azureuser]# su - oracle

Last login: Tue Feb 6 16:26:28 UTC 2024 on pts/0

[oracle@ora-02 ~]$ uname -a

Linux ora-02 4.18.0-372.9.1.el8.x86_64 #1 SMP Fri Apr 15 22:12:19 EDT 2022 x86_64 x86_64 x86_64 GNU/Linux

[oracle@ora-02 ~]$ df -h

Filesystem Size Used Avail Use% Mounted on
devtmpfs 7.7G 0 7.7G 0% /dev
tmpfs 7.8G 0 7.8G 0% /dev/shm
tmpfs 7.8G 49M 7.7G 1% /run
tmpfs 7.8G 0 7.8G 0% /sys/fs/cgroup
/dev/mapper/rootvg-rootlv 22G 17G 5.6G 75% /
/dev/mapper/rootvg-usrlv 10G 2.0G 8.1G 20% /usr
/dev/mapper/rootvg-homelv 1014M 40M 975M 4% /home
/dev/sda1 496M 106M 390M 22% /boot
/dev/mapper/rootvg-varlv 8.0G 958M 7.1G 12% /var
/dev/sda15 495M 5.9M 489M 2% /boot/efi
/dev/mapper/rootvg-tmplv 12G 8.4G 3.7G 70% /tmp
tmpfs 1.6G 0 1.6G 0% /run/user/54321
172.30.136.68:/ora-02-u03 250G 2.1G 248G 1% /u03
172.30.136.68:/ora-02-u01 100G 10G 91G 10% /u01
172.30.136.68:/ora-02-u02 250G 7.5G 243G 3% /u02
tmpfs 1.6G 0 1.6G 0% /run/user/1000
tmpfs 1.6G 0 1.6G 0% /run/user/0
172.30.136.68:/ora-01-u02-Clone-020624161543077 250G 8.2G 242G
[oracle@ora-02 ~]$ cat /etc/oratab
#
# This file is used by ORACLE utilities. It is created by root.sh
# and updated by either Database Configuration Assistant while creating
# a database or ASM Configuration Assistant while creating ASM
# instance.
#
# A colon, ':', is used as the field terminator. A new line terminates
# the entry. Lines beginning with a pound sign, '#', are comments.
#
# Entries are of the form:
#   $ORACLE_SID:$ORACLE_HOME:<N|Y>:  
# The first and second fields are the system identifier and home
# directory of the database respectively. The third field indicates
# to the dbstart utility that the database should, "Y", or should not,
# "N", be brought up at system boot time.
#
# Multiple entries with the same $ORACLE_SID are not allowed.
#
# NTAP2:/u01/app/oracle/product/19.0.0/NTAP2:Y
# SnapCenter Plug-in for Oracle Database generated entry (DO NOT
# REMOVE THIS LINE)
ntap1dev:/u01/app/oracle/product/19.0.0/NTAP2:N

[oracle@ora-02 ~]$ export ORACLE_SID=ntap1dev
[oracle@ora-02 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Tue Feb 6 16:29:02 2024
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0
SQL> select name, open_mode, log_mode from v$database;

<table>
<thead>
<tr>
<th>NAME</th>
<th>OPEN_MODE</th>
<th>LOG_MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTAP1DEV</td>
<td>READ WRITE</td>
<td>ARCHIVELOG</td>
</tr>
</tbody>
</table>

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.

Total System Global Area 3221223168 bytes
Fixed Size 9168640 bytes
Variable Size 654311424 bytes
Database Buffers 2550136832 bytes
Redo Buffers 7606272 bytes
Database mounted.

SQL> alter database noarchivelog;
Database altered.

SQL> alter database open;
Database altered.

SQL> select name, open_mode, log_mode from v$database;

<table>
<thead>
<tr>
<th>NAME</th>
<th>OPEN_MODE</th>
<th>LOG_MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTAP1DEV</td>
<td>READ WRITE</td>
<td>NOARCHIVELOG</td>
</tr>
</tbody>
</table>

SQL> show pdbs

<table>
<thead>
<tr>
<th>CON_ID</th>
<th>CON_NAME</th>
<th>OPEN_MODE</th>
<th>RESTRICTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>READ ONLY</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>NTAP1_PDB1</td>
<td>MOUNTED</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>NTAP1_PDB2</td>
<td>MOUNTED</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>NTAP1_PDB3</td>
<td>MOUNTED</td>
<td></td>
</tr>
</tbody>
</table>

SQL> alter pluggable database all open;
Where to find additional information

To learn more about the information described in this document, review the following documents and/or websites:

- Azure NetApp Files

- SnapCenter Software documentation
  https://docs.netapp.com/us-en/snapcenter/index.html

- TR-4987: Simplified, Automated Oracle Deployment on Azure NetApp Files with NFS

TR-4977: Oracle Database backup, restore and clone with SnapCenter Services - Azure

Allen Cao, Niyaz Mohamed, NetApp

Purpose

SnapCenter Services is the SaaS version of the classic SnapCenter database management UI tool that is available through the NetApp BlueXP cloud management console. It is an integral part of the NetApp cloud-backup, data-protection offering for databases such as Oracle and HANA running on Azure NetApp Files. This SaaS-based service simplifies traditional SnapCenter standalone server deployment that generally requires a Windows server operating in a Windows domain environment.

In this documentation, we demonstrate how you can set up SnapCenter Services to backup, restore, and clone Oracle databases deployed on Azure NetApp Files volumes and Azure compute instances. It is very easy to setup data protection for Oracle database deployed on Azure NetApp Files with web based BlueXP user interface.

This solution addresses the following use cases:

- Database backup with snapshots for Oracle databases hosted in Azure NetApp Files and Azure VMs
- Oracle database recovery in the case of a failure
- Fast cloning of primary databases for dev, test environments or other use cases

Audience

This solution is intended for the following audiences:

- The DBA who manages Oracle databases running on Azure NetApp Files storage
- The solution architect who is interested in testing Oracle database backup, restore, and clone in Azure
- The storage administrator who supports and manages the Azure NetApp Files storage
- The application owner who owns applications that are deployed to Azure NetApp Files storage and Azure VMs
Solution test and validation environment

The testing and validation of this solution was performed in a lab environment that might not match the final deployment environment. For more information, see the section [Key Factors for Deployment Consideration].

Architecture

This image provides a detailed picture of BlueXP backup and recovery for applications within the BlueXP console, including the UI, the connector, and the resources it manages.

Hardware and software components

Hardware

<table>
<thead>
<tr>
<th>Azure NetApp Files storage</th>
<th>Premium Service level</th>
<th>Auto QoS type, and 4TB in storage capacity in testing</th>
</tr>
</thead>
</table>

| Azure instance for compute | Standard B4ms (4 vcpus, 16 GiB memory) | Two instances deployed, one as primary DB server and the other as clone DB server |

Software

<table>
<thead>
<tr>
<th>RedHat Linux</th>
<th>Red Hat Enterprise Linux 8.7 (LVM) - x64 Gen2</th>
<th>Deployed RedHat subscription for testing</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Oracle Database</th>
<th>Version 19.18</th>
<th>Applied RU patch p34765931_190000_Linux-x86-64.zip</th>
</tr>
</thead>
</table>

| Oracle OPatch | Version 12.2.0.1.36 | Latest patch p6880880_190000_Linux-x86-64.zip |

| SnapCenter Service | Version v2.5.0-2822 | Agent Version v2.5.0-2822 |
Key factors for deployment consideration

• **Connector to be deployed in the same virtual network / subnet as databases and Azure NetApp Files.** When possible, the connector should be deployed in the same Azure virtual networks and resource groups, which enables connectivity to the Azure NetApp Files storage and the Azure compute instances.

• **An Azure user account or Active Directory service principle created at Azure portal for SnapCenter connector.** Deploying a BlueXP Connector requires specific permissions to create and configure a virtual machine and other compute resources, to configure networking, and to get access to the Azure subscription. It also requires permissions to later create roles and permissions for the Connector to operate. Create a custom role in Azure with permissions and assign to the user account or service principle. Review the following link for details: [Set up Azure permissions](#).

• **A ssh key pair created in the Azure resource group.** The ssh key pair is assigned to the Azure VM user for logging into the connector host and also the database VM host for deploying and executing a plug-in. BlueXP console UI uses the ssh key to deploy SnapCenter service plugin to database host for one-step plugin installation and application host database discovery.

• **A credential added to the BlueXP console setting.** To add Azure NetApp Files storage to the BlueXP working environment, a credential that grants permissions to access Azure NetApp Files from the BlueXP console needs to be set up in the BlueXP console setting.

• **java-11-openjdk installed on the Azure VM database instance host.** SnapCenter service installation requires java version 11. It needs to be installed on application host before plugin deployment attempt.

Solution deployment

There is extensive NetApp documentation with a broader scope to help you protect your cloud-native application data. The goal of this documentation is to provide step-by-step procedures that cover SnapCenter Service deployment with the BlueXP console to protect your Oracle database deployed on an Azure NetApp Files storage and an Azure compute instance.

To get started, complete the following steps:

• Read the general instructions [Protect your cloud native applications data](#) and the sections related to Oracle and Azure NetApp Files.

• Watch the following video walkthrough

  Video of deployment of Oracle and ANF

Prerequisites for SnapCenter service deployment

Deployment requires the following prerequisites.

1. A primary Oracle database server on an Azure VM instance with an Oracle database fully deployed and running.

2. An Azure NetApp Files storage service capacity pool deployed in Azure that has capacity to meet the database storage needs listed in hardware component section.

3. A secondary database server on an Azure VM instance that can be used for testing the cloning of an Oracle database to an alternate host for the purpose of supporting a dev/test workload or any use cases that requires a full data set of production Oracle database.

4. For additional information for Oracle database deployment on Azure NetApp Files and Azure compute instance, see [Oracle Database Deployment and Protection on Azure NetApp Files](#).
Onboarding to BlueXP preparation

1. Use the link NetApp BlueXP to sign up for BlueXP console access.

2. Create an Azure user account or an Active Directory service principle and grant permissions with role in Azure portal for Azure connector deployment.

3. To set up BlueXP to manage Azure resources, add a BlueXP credential with details of an Active Directory service principal that BlueXP can use to authenticate with Azure Active Directory (App client ID), a client secret for the service principal application (Client Secret), and the Active Directory ID for your organization (Tenant ID).

4. You also need the Azure virtual network, resources group, security group, an SSH key for VM access, etc. ready for connector provisioning and database plugin installation.

Deploy a connector for SnapCenter services
1. Login to the BlueXP console.

2. Click on **Connector** drop down arrow and **Add Connector** to launch the connector provisioning workflow.

3. Choose your cloud provider (in this case, **Microsoft Azure**).
4. Skip the Permission, Authentication, and Networking steps if you already have them set up in your Azure account. If not, you must configure these before proceeding. From here, you could also retrieve the permissions for the Azure policy that is referenced in the previous section "Onboarding to BlueXP preparation."
5. Click on **Skip to Deployment** to configure your connector **Virtual Machine Authentication**. Add the SSH key pair you have created in Azure resource group during onboarding to BlueXP preparation for connector OS authentication.
6. Provide a name for the connector instance, select **Create** and accept default **Role Name** under **Details**, and choose the subscription for the Azure account.
7. Configure networking with the proper **VNet, Subnet**, and disable **Public IP** but ensure that the connector has the internet access in your Azure environment.
8. Configure the **Security Group** for the connector that allows HTTP, HTTPS, and SSH access.

9. Review the summary page and click **Add** to start connector creation. It generally takes about 10 mins to complete deployment. Once completed, the connector instance VM appears in the Azure portal.
10. After the connector is deployed, the newly created connector appears under **Connector** drop-down.
Define a credential in BlueXP for Azure resources access
1. Click on setting icon on top right corner of BlueXP console to open **Account credentials** page, click **Add credentials** to start credential configuration workflow.

2. Choose credential location as - **Microsoft Azure - BlueXP**.

3. Define Azure credentials with proper **Client Secret**, **Client ID**, and **Tenant ID**, which should have been gathered during previous BlueXP onboarding process.
4. Review and Add.

5. You may also need to associate a **Marketplace Subscription** with the credential.
SnapCenter services setup
With the Azure credential configured, SnapCenter services can now be set up with the following procedures:


2. Choose Microsoft Azure as the location and click on Discover.

3. Name Working Environment and choose Credential Name created in previous section, and click Continue.
4. BlueXP console returns to **My working environments** and discovered Azure NetApp Files from Azure now appears on **Canvas**.

5. Click on **Azure NetApp Files** icon, then **Enter Working Environment** to view Oracle database volumes deployed in Azure NetApp Files storage.
6. From the left-hand sidebar of the console, hover your mouse over the protection icon, and then click Protection > Applications to open the Applications launch page. Click Discover Applications.

7. Select Cloud Native as the application source type.
8. Choose **Oracle** for the application type, click on **Next** to open host details page.

9. Select **Using SSH** and provide the Oracle Azure VM details such as **IP address**, **Connector**, Azure VM management **Username** such as azureuser. Click on **Add SSH Private Key** to paste in the SSH key pair that you used to deploy the Oracle Azure VM. You will also be prompted to confirm the fingerprint.
10. Move on to next **Configuration** page to setup sudoer access on Oracle Azure VM.
11. Review and click on Discover Applications to install a plugin on the Oracle Azure VM and discover Oracle database on the VM in one step.

12. Discovered Oracle databases on Azure VM are added to Applications, and the Applications page lists the number of hosts and Oracle databases within the environment. The database Protection Status initially shows as Unprotected.
This completes the initial setup of SnapCenter services for Oracle. The next three sections of this document describe Oracle database backup, restore, and clone operations.

**Oracle database backup**
1. Our test Oracle database in Azure VM is configured with three volumes with an aggregate total storage about 1.6 TiB. This gives context about the timing for the snapshot backup, restore, and clone of a database of this size.

```
[oracle@acao-ora01 ~]$ df -h
Filesystem                 Size  Used Avail Use% Mounted on
devtmpfs                   7.9G     0  7.9G   0% /dev
tmpfs                      7.9G     0  7.9G   0% /dev/shm
tmpfs                      7.9G  17M  7.9G   1% /run
tmpfs                      7.9G     0  7.9G   0% /sys/fs/cgroup
/dev/mapper/rootvg-rootlv   40G   23G   15G  62% /
/dev/mapper/rootvg-usrlv   9.8G  1.6G  7.7G  18% /usr
/dev/sda2                  496M  115M  381M  24% /boot
/dev/mapper/rootvg-varlv   7.9G  787M  6.7G  11% /var
/dev/mapper/rootvg-homelv  976M  323M  586M  36% /home
/dev/mapper/rootvg-optlv   2.0G   9.6M  1.8G   1% /opt
/dev/mapper/rootvg-tmplv   2.0G   22M  1.8G   2% /tmp
/dev/sdal                  500M   6.8M  493M   2% /boot/efi
172.30.136.68:/ora01-u01  100G  23G   78G  23% /u01
172.30.136.68:/ora01-u03  500G  117G  384G  24% /u03
172.30.136.68:/ora01-u02 1000G  804G  197G  81% /u02
tmpfs                      1.6G     0  1.6G   0% /run/user/1000
[oracle@acao-ora01 ~]$
```

1. To protect database, click the three dots next to the database **Protection Status**, and then click **Assign Policy** to view the default preloaded or user defined database protection policies that can be applied to your Oracle databases. Under **Settings - Policies**, you have option to create your own policy with a customized backup frequency and backup data-retention window.
2. When you are happy with the policy configuration, you can then **Assign** your policy of choice to protect the database.

3. After the policy is applied, the database protection status changed to **Protected** with a green check mark. BlueXP executes the snapshot backup according to the schedule defined. In addition, **ON-Demand Backup** is available from the three-dot drop down menu as shown below.
4. From **Job Monitoring** tab, backup job details can be viewed. Our test results showed that it took about 4 minutes to backup an Oracle database about 1.6 TiB.

5. From three-dot drop down menu **View Details**, you can view the backup sets created from snapshot backup.
6. Database backup details include the **Backup Name**, **Backup Type**, **SCN**, **RMAN Catalog**, and **Backup Time**. A backup set contains application-consistent snapshots for data volume and log volume respectively. A log volume snapshot takes place right after a database data volume snapshot. You could apply a filter if you are looking for a particular backup in the backup list.

---

**Oracle database restore and recovery**
1. For a database restore, click the three-dot drop down menu for the particular database to be restored in Applications, then click Restore to initiate database restore and recovery workflow.

2. Choose your Restore Point by time stamp. Each time stamp in the list represents an available database backup set.

3. Choose your Restore Location to original location for an Oracle database in place restore and recovery.
4. Define your **Restore Scope**, and **Recovery Scope**. All Logs mean a full recovery up to date including current logs.

5. Review and **Restore** to start database restore and recovery.
6. From the **Job Monitoring** tab, we observed that it took 2 minutes to run a full database restore and recovery up to date.
Database clone procedures are similar to restore but to an alternate Azure VM with identical Oracle software stack pre-installed and configured.

Ensure that your Azure NetApp File storage has sufficient capacity for a cloned database the same size as the primary database to be cloned. The alternate Azure VM has been added to Applications.

1. Click the three-dot drop down menu for the particular database to be cloned in Applications, then click **Restore** to initiate clone workflow.

2. Select the **Restore Point** and check the **Restore to alternate location**.
3. In the next **Configuration** page, set alternate **Host**, new database **SID**, and **Oracle Home** as configured at alternate Azure VM.

4. Review **General** page shows the details of cloned database such as **SID**, alternate host, data file locations, recovery scope etc.

5. Review **Database parameters** page shows the details of cloned database configuration as well as some database parameters setting.
6. Monitor the cloning job status from the **Job Monitoring** tab, we observed that it took 8 minutes to clone a 1.6 TiB Oracle database.

7. Validate the cloned database in **BlueXP Applications** page that showed the cloned database was immediately registered with **BlueXP**.
8. Validate the cloned database on the Oracle Azure VM that showed the cloned database was running as expected.
This completes the demonstration of an Oracle database backup, restore, and clone in Azure with NetApp BlueXP console using SnapCenter Service.

**Additional information**

To learn more about the information that is described in this document, review the following documents and/or websites:

- Set up and administer BlueXP
  

- BlueXP backup and recovery documentation
  

- Azure NetApp Files
TR-4964: Oracle Database backup, restore and clone with SnapCenter Services - AWS

Allen Cao, Niyaz Mohamed, NetApp

Purpose

SnapCenter Services is the SaaS version of the classic SnapCenter database management UI tool that is available through the NetApp BlueXP cloud management console. It is an integral part of the NetApp cloud-backup, data-protection offering for databases such as Oracle and HANA running on NetApp cloud storage. This SaaS-based service simplifies traditional SnapCenter standalone server deployment that generally requires a Windows server operating in a Windows domain environment.

In this documentation, we demonstrate how you can set up SnapCenter Services to backup, restore, and clone Oracle databases deployed to Amazon FSx for ONTAP storage and EC2 compute instances. Although it is much easier to set up and use, SnapCenter Services deliver key functionalities that are available in the legacy SnapCenter UI tool.

This solution addresses the following use cases:

- Database backup with snapshots for Oracle databases hosted in Amazon FSx for ONTAP
- Oracle database recovery in the case of a failure
- Fast and storage-efficient cloning of primary databases for a dev/test environment or other use cases

Audience

This solution is intended for the following audiences:

- The DBA who manages Oracle databases running on Amazon FSx for ONTAP storage
- The solution architect who is interested in testing Oracle database backup, restore, and clone in the public AWS cloud
- The storage administrator who supports and manages the Amazon FSx for ONTAP storage
- The application owner who owns applications that are deployed to Amazon FSx for ONTAP storage

Solution test and validation environment

The testing and validation of this solution was performed in an AWS FSx and EC2 environment that might not match the final deployment environment. For more information, see the section [Key Factors for Deployment Consideration].

Architecture
This image provides a detailed picture of BlueXP backup and recovery for applications within the BlueXP console, including the UI, the connector, and the resources it manages.

**Hardware and software components**

**Hardware**

<table>
<thead>
<tr>
<th>Component</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSx ONTAP storage</td>
<td>Current version offered by AWS</td>
</tr>
<tr>
<td></td>
<td>One FSx HA cluster in the same VPC and availability zone</td>
</tr>
<tr>
<td>EC2 instance for compute</td>
<td>t2.xlarge/4vCPU/16G</td>
</tr>
<tr>
<td></td>
<td>Two EC2 T2 xlarge EC2 instances, one as primary DB server and the other as clone DB server</td>
</tr>
</tbody>
</table>

**Software**

<table>
<thead>
<tr>
<th>Component</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>RedHat Linux</td>
<td>RHEL-8.6.0_HVM-20220503-x86_64-2-Hourly2-GP2</td>
</tr>
<tr>
<td></td>
<td>Deployed RedHat subscription for testing</td>
</tr>
<tr>
<td>Oracle Grid Infrastructure</td>
<td>Version 19.18</td>
</tr>
<tr>
<td></td>
<td>Applied RU patch p34762026_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle Database</td>
<td>Version 19.18</td>
</tr>
<tr>
<td></td>
<td>Applied RU patch p34765931_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>Oracle OPatch</td>
<td>Version 12.2.0.1.36</td>
</tr>
<tr>
<td></td>
<td>Latest patch p6880880_190000_Linux-x86-64.zip</td>
</tr>
<tr>
<td>SnapCenter Service</td>
<td>Version v2.3.1.2324</td>
</tr>
</tbody>
</table>
Key factors for deployment consideration

- **Connector to be deployed in the same VPC as database and FSx.** When possible, the connector should be deployed in the same AWS VPC, which enables connectivity to the FSx storage and the EC2 compute instance.

- **An AWS IAM policy created for SnapCenter connector.** The policy in JSON format is available in the detailed SnapCenter service documentation. When you launch connector deployment with the BlueXP console, you are also prompted to set up the prerequisites with details of required permission in JSON format. The policy should be assigned to the AWS user account that owns the connector.

- **The AWS account access key and the SSH key pair created in the AWS account.** The SSH key pair is assigned to the ec2-user for logging into the connector host and then deploying a database plug-in to the EC2 DB server host. The access key grants permission for provisioning the required connector with IAM policy above.

- **A credential added to the BlueXP console setting.** To add Amazon FSx for ONTAP to the BlueXP working environment, a credential that grants BlueXP permissions to access Amazon FSx for ONTAP is set up in the BlueXP console setting.

- **java-11-openjdk installed on the EC2 database instance host.** SnapCenter service installation requires java version 11. It needs to be installed on application host before plugin deployment attempt.

Solution deployment

There is extensive NetApp documentation with a broader scope to help you protect your cloud-native application data. The goal of this documentation is to provide step-by-step procedures that cover SnapCenter Service deployment with the BlueXP console to protect your Oracle database deployed to Amazon FSx for ONTAP and an EC2 compute instance. This document fills in certain details that might be missing from more general instructions.

To get started, complete the following steps:

- Read the general instructions [Protect your cloud native applications data](#) and the sections related to Oracle and Amazon FSx for ONTAP.

- Watch the following video walkthrough.

Solution Deployment

Prerequisites for SnapCenter service deployment

Deployment requires the following prerequisites.

1. A primary Oracle database server on an EC2 instance with an Oracle database fully deployed and running.

2. An Amazon FSx for ONTAP cluster deployed in AWS that is hosting the database volumes above.

3. An optional database server on an EC2 instance that can be used for testing the cloning of an Oracle database to an alternate host for the purpose of supporting a dev/test workload or any use cases that requires a full data set of a production Oracle database.

4. If you need help to meet the above prerequisites for Oracle database deployment on Amazon FSx for ONTAP and EC2 compute instance, see [Oracle Database Deployment and Protection in AWS FSx/EC2 with iSCSI/ASM](#) or white paper [Oracle Database Deployment on EC2 and FSx Best Practices](#)
Onboarding to BlueXP preparation

1. Use the link [NetApp BlueXP](#) to sign up for BlueXP console access.

2. Login to your AWS account to create an IAM policy with proper permissions and assign the policy to the AWS account that will be used for BlueXP connector deployment.

   ![IAM Policy Configuration](image)

   The policy should be configured with a JSON string that is available in NetApp documentation. The JSON string can also be retrieved from the page when connector provisioning is launched and you are prompted for the prerequisites permissions assignment.

3. You also need the AWS VPC, subnet, security group, an AWS user account access key and secrets, an SSH key for ec2-user, and so on ready for connector provisioning.

Deploy a connector for SnapCenter services
1. Login to the BlueXP console. For a shared account, it is a best practice to create an individual workspace by clicking Account > Manage Account > Workspace to add a new workspace.

2. Click Add a Connector to launch the connector provisioning workflow.
1. Choose your cloud provider (in this case, Amazon Web Services).

1. Skip the **Permission**, **Authentication**, and **Networking** steps if you already have them set up in your AWS account. If not, you must configure these before proceeding. From here, you could also retrieve the permissions for the AWS policy that is referenced in the previous section "Onboarding to BlueXP preparation."
1. Enter your AWS account authentication with **Access Key** and **Secret Key**.
2. Name the connector instance and select **Create Role** under **Details**.

1. Configure networking with the proper **VPC**, **Subnet**, and **SSH Key Pair** for connector access.
2. Set the **Security Group** for the connector.
3. Review the summary page and click **Add** to start connector creation. It generally takes about 10 mins to complete deployment. Once completed, the connector instance appears in the AWS EC2 dashboard.

Define a credential in BlueXP for AWS resources access
1. First, from AWS EC2 console, create a role in **Identity and Access Management (IAM)** menu **Roles**, **Create role** to start role creation workflow.

2. In **Select trusted entity** page, choose **AWS account**, **Another AWS account**, and paste in the BlueXP account ID, which can be retrieved from BlueXP console.

3. Filter permission policies by fsx and add **Permissions policies** to the role.
4. In Role details page, name the role, add a description, then click Create role.

5. Back to BlueXP console, click on setting icon on top right corner of the console to open Account credentials page, click Add credentials to start credential configuration workflow.

6. Choose credential location as - Amazon Web Services - BlueXP.
7. Define AWS credentials with proper **Role ARN**, which can be retrieved from AWS IAM role created in step one above. BlueXP **account ID**, which is used for creating AWS IAM role in step one.

8. Review and **Add**.
SnapCenter services setup
With the connector deployed and the credential added, SnapCenter services can now be set up with the following procedure:

1. From **My Working Environment** click **Add working Environment** to discover FSx deployed in AWS.

1. Choose **Amazon Web Services** as the location.

1. Click **Discover Existing** next to **Amazon FSx for ONTAP**.
1. Select the **Credentials Name** that you have created in previous section to grant BlueXP with the permissions that it needs to manage FSx for ONTAP. If you have not added credentials, you can add it from the **Settings** menu at the top right corner of the BlueXP console.

2. Choose the AWS region where Amazon FSx for ONTAP is deployed, select the FSx cluster that is hosting the Oracle database and click Add.
1. The discovered Amazon FSx for ONTAP instance now appears in the working environment.

1. You can log into the FSx cluster with your fsxadmin account credentials.
1. After you log into Amazon FSx for ONTAP, review your database storage information (such as database volumes).

1. From the left-hand sidebar of the console, hover your mouse over the protection icon, and then click **Protection > Applications** to open the Applications launch page. Click **Discover Applications**.
1. Select **Cloud Native** as the application source type.

1. Choose **Oracle** for the application type.
1. Fill in the AWS EC2 Oracle application host details. Choose **Using SSH** as **Host Installation Type** for one step plugin installation and database discovery. Then, click on **Add SSH Private Key**.

2. Paste in your ec2-user SSH key for the database EC2 host and click on **Validate** to proceed.
3. You will be prompted for **Validating fingerprint** to proceed.

4. Click on **Next** to install an Oracle database plugin and discover the Oracle databases on the EC2 host. Discovered databases are added to **Applications**. The database **Protection Status** shows as **Unprotected** when initially discovered.
This completes the initial setup of SnapCenter services for Oracle. The next three sections of this document describe Oracle database backup, restore, and clone operations.

**Oracle database backup**
1. Click the three dots next to the database **Protection Status**, and then click **Policies** to view the default preloaded database protection policies that can be applied to protect your Oracle databases.

1. You can also create your own policy with a customized backup frequency and backup data-retention window.

1. When you are happy with the policy configuration, you can then assign your policy of choice to protect the database.
1. Choose the policy to assign to the database.

1. After the policy is applied, the database protection status changed to **Protected** with a green check mark.
1. The database backup runs on a predefined schedule. You can also run a one-off on-demand backup as shown below.

1. The database backups details can be viewed by clicking **View Details** from the menu list. This includes the backup name, backup type, SCN, and backup date. A backup set covers a snapshot for both data volume and log volume. A log volume snapshot takes place right after a database volume snapshot. You can apply a filter if you are looking for a particular backup in a long list.
Oracle database restore and recovery
1. For a database restore, choose the right backup, either by the SCN or backup time. Click the three dots from the database data backup, and then click **Restore** to initiate database restore and recovery.

1. Choose your restore setting. If you are sure that nothing has changed in the physical database structure after the backup (such as the addition of a data file or a disk group), you can use the **Force in place restore** option, which is generally faster. Otherwise, do not check this box.

1. Review and start database restore and recovery.
1. From the **Job Monitoring** tab, you can view the status of the restore job as well as any details while it is running.
Oracle database clone
To clone a database, launch the clone workflow from the same database backup details page.

1. Select the right database backup copy, click the three dots to view the menu, and choose the **Clone** option.

2. Select the **Basic** option if you don’t need to change any cloned database parameters.

3. Alternatively, select **Specification file**, which gives you the option of downloading the current init file, making changes, and then uploading it back to the job.
1. Review and launch the job.

1. Monitor the cloning job status from the **Job Monitoring** tab.
1. Validate the cloned database on the EC2 instance host.

<table>
<thead>
<tr>
<th>Job Name</th>
<th>Job ID</th>
<th>Start Time</th>
<th>End Time</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloning Oracle Database</td>
<td>cc3baa4fe-012-4c52-ab3b-40856ba293d</td>
<td>Mar 24, 2023, 1:30:01 pm</td>
<td>--</td>
<td>0 seconds</td>
</tr>
<tr>
<td>Running pre scripts</td>
<td>SF53a5e-413e-4c52-ab3b-40856ba293d</td>
<td>Mar 24, 2023, 1:30:01 pm</td>
<td>Mar 24, 2023, 1:30:01 pm</td>
<td>0 seconds</td>
</tr>
<tr>
<td>Validating clone request</td>
<td>R5ab6a4.2ab3-4.5e-9c7c</td>
<td>Mar 24, 2023, 1:30:01 pm</td>
<td>Mar 24, 2023, 1:30:02 pm</td>
<td>7 seconds</td>
</tr>
</tbody>
</table>
Multiple entries with the same ORACLE_SID are not allowed.

+ASM:/u01/app/oracle/product/19.0.0/grid:N
db1:/u01/app/oracle/product/19.0.0/db1:N

SnapCenter Plug-in for Oracle Database generated entry (DO NOT REMOVE THIS LINE)
dbcloner:/u01/app/oracle/product/19.0.0/db1:N

[oracle@ip-172-30-15-58 ~]$ oraclic stat res -t

<table>
<thead>
<tr>
<th>Name</th>
<th>Target</th>
<th>State</th>
<th>Server</th>
<th>State details</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora.DATAG</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.DATAG/dbcloner</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.LISTENER.1snr</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.LOGS</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.LOGS_SCO_27481385658</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.sm</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
<td>Started,STABLE</td>
</tr>
<tr>
<td>ora.sm</td>
<td>OFFLINE</td>
<td>OFFLINE</td>
<td>ip-172-30-15-58</td>
<td>STABLE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cluster Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora.css</td>
</tr>
<tr>
<td>ora.db1.db</td>
</tr>
<tr>
<td>ora.dbcloner.db</td>
</tr>
<tr>
<td>ora.diskmon</td>
</tr>
<tr>
<td>ora.driver.sdf</td>
</tr>
<tr>
<td>ora.evmd</td>
</tr>
</tbody>
</table>

[oracle@ip-172-30-15-58 ~]$ export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db1
[oracle@ip-172-30-15-58 ~]$ export ORACLE_SID=dbcloner
[oracle@ip-172-30-15-58 ~]$ export PATH=$ORACLE_HOME/bin:$PATH
[oracle@ip-172-30-15-58 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Mar 24 18:32:21 2023
Version 19.0.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.0.0.0.0

SQL> select name, open_mode from v$database;

NAME     OPEN_MODE
--------   ------------
DBCLONE    READ WRITE

SQL>
Hybrid Cloud Database Solutions with SnapCenter

TR-4908: Hybrid Cloud Database Solutions with SnapCenter Overview

Alan Cao, Felix Melligan, NetApp

This solution provides NetApp field and customers with instructions and guidance for configuring, operating, and migrating databases to a hybrid cloud environment using the NetApp SnapCenter GUI-based tool and the NetApp storage service CVO in public clouds for the following use cases:

- Database dev/test operations in the hybrid cloud
- Database disaster recovery in the hybrid cloud

Today, many enterprise databases still reside in private corporate data centers for performance, security, and/or other reasons. This hybrid cloud database solution enables enterprises to operate their primary databases on site while using a public cloud for dev/test database operations as well as for disaster recovery to reduce licensing and operational costs.

Many enterprise databases, such as Oracle, SQL Server, SAP HANA, and so on, carry high licensing and operational costs. Many customers pay a one-time license fee as well as annual support costs based on the number of compute cores in their database environment, whether the cores are used for development, testing, production, or disaster recovery. Many of those environments might not be fully utilized throughout the application lifecycle.

The solutions provide an option for customers to potentially reduce their licensable cores count by moving their database environments devoted to development, testing, or disaster recovery to the cloud. By using public-cloud scale, redundancy, high availability, and a consumption-based billing model, the cost saving for licensing and operation can be substantial, while not sacrificing any application usability or availability.
Beyond potential database license-cost savings, the NetApp capacity-based CVO license model allows customers to save storage costs on a per-GB basis while empowering them with high level of database manageability that is not available from competing storage services. The following chart shows a storage cost comparison of popular storage services available in the public cloud.

![Storage Cost Comparison Chart]

This solution demonstrates that, by using the SnapCenter GUI-based software tool and NetApp SnapMirror technology, hybrid cloud database operations can be easily setup, implemented, and operated.

The following videos demonstrate SnapCenter in action:

- Backup of an Oracle database across a Hybrid Cloud using SnapCenter
- SnapCenter- Clone DEV/TEST to AWS Cloud for an Oracle database

Notably, although the illustrations throughout this document show CVO as a target storage instance in the public cloud, the solution is also fully validated for the new release of the FSx ONTAP storage engine for AWS.

To test drive the solution and use cases for yourself, a NetApp Lab-on-Demand SL10680 can be requested at following xref: /

**Solution Architecture**

The following architecture diagram illustrates a typical implementation of enterprise database operation in a hybrid cloud for dev/test and disaster recovery operations.
In normal business operations, synchronized database volumes in the cloud can be cloned and mounted to dev/test database instances for applications development or testing. In the event of a failure, the synchronized database volumes in the cloud can then be activated for disaster recovery.

**SnapCenter Requirements**

This solution is designed in a hybrid cloud setting to support on-premises production databases that can burst to all of the popular public clouds for dev/test and disaster recovery operations.

This solution supports all databases that are currently supported by SnapCenter, although only Oracle and SQL Server databases are demonstrated here. This solution is validated with virtualized database workloads, although bare-metal workloads are also supported.

We assume that production database servers are hosted on-premises with DB volumes presented to DB hosts from a ONTAP storage cluster. SnapCenter software is installed on-premises for database backup and data replication to the cloud. An Ansible controller is recommended but not required for database deployment automation or OS kernel and DB configuration syncing with a standby DR instance or dev/test instances in the public cloud.

**Requirements**
<table>
<thead>
<tr>
<th>Environment</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-premises</td>
<td>Any databases and versions supported by SnapCenter</td>
</tr>
<tr>
<td></td>
<td>SnapCenter v4.4 or higher</td>
</tr>
<tr>
<td></td>
<td>Ansible v2.09 or higher</td>
</tr>
<tr>
<td></td>
<td>ONTAP cluster 9.x</td>
</tr>
<tr>
<td></td>
<td>Intercluster LIFs configured</td>
</tr>
<tr>
<td></td>
<td>Connectivity from on-premises to a cloud VPC (VPN, interconnect, and so on)</td>
</tr>
<tr>
<td></td>
<td>Networking ports open</td>
</tr>
<tr>
<td></td>
<td>- ssh 22</td>
</tr>
<tr>
<td></td>
<td>- tcp 8145, 8146, 10000, 11104, 11105</td>
</tr>
<tr>
<td>Cloud - AWS</td>
<td>Cloud Manager Connector</td>
</tr>
<tr>
<td></td>
<td>Cloud Volumes ONTAP</td>
</tr>
<tr>
<td></td>
<td>Matching DB OS EC2 instances to On-prem</td>
</tr>
<tr>
<td>Cloud - Azure</td>
<td>Cloud Manager Connector</td>
</tr>
<tr>
<td></td>
<td>Cloud Volumes ONTAP</td>
</tr>
<tr>
<td></td>
<td>Matching DB OS Azure Virtual Machines to On-prem</td>
</tr>
<tr>
<td>Cloud - GCP</td>
<td>Cloud Manager Connector</td>
</tr>
<tr>
<td></td>
<td>Cloud Volumes ONTAP</td>
</tr>
<tr>
<td></td>
<td>Matching DB OS Google Compute Engine instances to on-premises</td>
</tr>
</tbody>
</table>

**Prerequisites configuration**

Certain prerequisites must be configured both on-premises and in the cloud before the execution of hybrid cloud database workloads. The following section provides a high-level summary of this process, and the following links provide further information about necessary system configuration.

**On premises**
- SnapCenter installation and configuration
- On-premises database server storage configuration
- Licensing requirements
- Networking and security
- Automation

**Public cloud**
- A NetApp Cloud Central login
- Network access from a web browser to several endpoints
- A network location for a connector
Important considerations:

1. Where to deploy the Cloud Manager Connector?
2. Cloud Volume ONTAP sizing and architecture
3. Single node or high availability?

The following links provide further details:

On Premises

Public Cloud

Prerequisites on-premises

The following tasks must be completed on-premises to prepare the SnapCenter hybrid-cloud database workload environment.

SnapCenter installation and configuration

The NetApp SnapCenter tool is a Windows-based application that typically runs in a Windows domain environment, although workgroup deployment is also possible. It is based on a multitiered architecture that includes a centralized management server (the SnapCenter server) and a SnapCenter plug-in on the database server hosts for database workloads. Here are a few key considerations for hybrid-cloud deployment.

- **Single instance or HA deployment.** HA deployment provides redundancy in the case of a single SnapCenter instance server failure.
- **Name resolution.** DNS must be configured on the SnapCenter server to resolve all database hosts as well as on the storage SVM for forward and reverse lookup. DNS must also be configured on database servers to resolve the SnapCenter server and the storage SVM for both forward and reverse lookup.
- **Role-based access control (RBAC) configuration.** For mixed database workloads, you might want to use RBAC to segregate management responsibility for different DB platform such as an admin for Oracle database or an admin for SQL Server. Necessary permissions must be granted for the DB admin user.
- **Enable policy-based backup strategy.** To enforce backup consistency and reliability.
- **Open necessary network ports on the firewall.** For the on-premises SnapCenter server to communicate with agents installed in the cloud DB host.
- **Ports must be open to allow SnapMirror traffic between on-prem and public cloud.** The SnapCenter server relies on ONTAP SnapMirror to replicate onsite Snapshot backups to cloud CVO storage SVMs.

After careful pre-installation planning and consideration, click this [SnapCenter installation workflow](#) for details of SnapCenter installation and configuration.

On-premises database server storage configuration

Storage performance plays an important role in the overall performance of databases and applications. A well-designed storage layout can not only improve DB performance but also make it easy to manage database backup and recovery. Several factors should be considered when defining your storage layout, including the size of the database, the rate of expected data change for the database, and the frequency with which you perform backups.
Directly attaching storage LUNs to the guest VM by either NFS or iSCSI for virtualized database workloads generally provides better performance than storage allocated via VMDK. NetApp recommends the storage layout for a large SQL Server database on LUNs depicted in the following figure.

![Storage Layout Diagram](image)

The following figure shows the NetApp recommended storage layout for small or medium SQL Server database on LUNs.

![Small/Medium Database Layout Diagram](image)

The Log directory is dedicated to SnapCenter to perform transaction log rollup for database recovery. For an extra large database, multiple LUNS can be allocated to a volume for better performance.

For Oracle database workloads, SnapCenter supports database environments backed by ONTAP storage that are mounted to the host as either physical or virtual devices. You can host the entire database on a single or multiple storage devices based on the criticality of the environment. Typically, customers isolate data files on dedicated storage from all other files such as control files, redo files, and archive log files. This helps administrators to quickly restore (ONTAP single-file SnapRestore) or clone a large critical database (petabyte scale) using Snapshot technology within few seconds to minutes.
For mission critical workloads that are sensitive to latency, a dedicated storage volume should be deployed to different types of Oracle files to achieve the best latency possible. For a large database, multiple LUNs (NetApp recommends up to eight) per volume should be allocated to data files.

![Datafile and Control file layout](image)

For smaller Oracle databases, SnapCenter supports shared storage layouts in which you can host multiple databases or part of a database on the same storage volume or LUN. As an example of this layout, you can host data files for all the databases on a +DATA ASM disk group or a volume group. The remainder of the files (redo, archive log, and control files) can be hosted on another dedicated disk group or volume group (LVM). Such a deployment scenario is illustrated below.

![Shared storage layout](image)

To facilitate the relocation of Oracle databases, the Oracle binary should be installed on a separate LUN that is included in the regular backup policy. This ensures that in the case of database relocation to a new server host, the Oracle stack can be started for recovery without any potential issues due to an out-of-sync Oracle binary.

### Licensing requirements

SnapCenter is licensed software from NetApp. It is generally included in an on-premises ONTAP license. However, for hybrid cloud deployment, a cloud license for SnapCenter is also required to add CVO to SnapCenter as a target data replication destination. Please review following links for SnapCenter standard capacity-based license for details:

[SnapCenter standard capacity-based licenses](#)

### Networking and security

In a hybrid database operation that requires an on-premises production database that is burstable to cloud for dev/test and disaster recovery, networking and security is important factor to consider when setting up the environment and connecting to the public cloud from an on-premises data center.

Public clouds typically use a virtual private cloud (VPC) to isolate different users within a public-cloud platform. Within an individual VPC, security is controlled using measures such as security groups that are configurable based on user needs for the lockdown of a VPC.

The connectivity from the on-premises data center to the VPC can be secured through a VPN tunnel. On the VPN gateway, security can be hardened using NAT and firewall rules that block attempts to establish network...
connections from hosts on the internet to hosts inside the corporate data center.

For networking and security considerations, review the relevant inbound and outbound CVO rules for your public cloud of choice:

- Security group rules for CVO - AWS
- Security group rules for CVO - Azure
- Firewall rules for CVO - GCP

**Using Ansible automation to sync DB instances between on-premises and the cloud - optional**

To simplify management of a hybrid-cloud database environment, NetApp highly recommends but does not require that you deploy an Ansible controller to automate some management tasks, such as keeping compute instances on-premises and in the cloud in sync. This is particularly important because an out-of-sync compute instance in the cloud might render the recovered database in the cloud error prone because of missing kernel packages and other issues.

The automation capability of an Ansible controller can also be used to augment SnapCenter for certain tasks, such as breaking up the SnapMirror instance to activate the DR data copy for production.

Follow these instructions to set up your Ansible control node for RedHat or CentOS machines: [RedHat/CentOS Ansible Controller Setup](#).
Follow these instructions to set up your Ansible control node for Ubuntu or Debian machines: [Ubuntu/Debian Ansible Controller Setup](#).

**Prerequisites for the public cloud**

Before we install the Cloud Manager connector and Cloud Volumes ONTAP and configure SnapMirror, we must perform some preparation for our cloud environment. This page describes the work that needs to be done as well as the considerations when deploying Cloud Volumes ONTAP.

**Cloud Manager and Cloud Volumes ONTAP deployment prerequisites checklist**

- A NetApp Cloud Central login
- Network access from a web browser to several endpoints
- A network location for a Connector
- Cloud provider permissions
- Networking for individual services

For more information about what you need to get started, visit our [cloud documentation](#).

**Considerations**

1. **What is a Cloud Manager connector?**

   In most cases, a Cloud Central account admin must deploy a connector in your cloud or on-premises network. The connector enables Cloud Manager to manage resources and processes within your public cloud environment.

   For more information about Connectors, visit our [cloud documentation](#).
2. Cloud Volumes ONTAP sizing and architecture

When deploying Cloud Volumes ONTAP, you are given the choice of either a predefined package or the creation of your own configuration. Although many of these values can be changed later on nondisruptively, there are some key decisions that need to be made before deployment based on the workloads to be deployed in the cloud.

Each cloud provider has different options for deployment and almost every workload has its own unique properties. NetApp has a CVO sizing tool that can help size deployments correctly based on capacity and performance, but it has been built around some basic concepts which are worth considering:

- Capacity required
- Network capability of the cloud virtual machine
- Performance characteristics of cloud storage

The key is to plan for a configuration that not only satisfies the current capacity and performance requirements, but also looks at future growth. This is generally known as capacity headroom and performance headroom.

If you would like further information, read the documentation about planning correctly for AWS, Azure, and GCP.

3. Single node or high availability?

In all clouds, there is the option to deploy CVO in either a single node or in a clustered high availability pair with two nodes. Depending on the use case, you might wish to deploy a single node to save costs or an HA pair to provide further availability and redundancy.

For a DR use case or spinning up temporary storage for development and testing, single nodes are common since the impact of a sudden zonal or infrastructure outage is lower. However, for any production use case, when the data is in only a single location, or when the dataset must have more redundancy and availability, high availability is recommended.

For further information about the architecture of each cloud’s version of high availability, visit the documentation for AWS, Azure and GCP.

Getting started overview

This section provides a summary of the tasks that must be completed to meet the prerequisite requirements as outlined in previous section. The following section provide a high level tasks list for both on-premises and public cloud operations. The detailed processes and procedures can be accessed by clicking on the relevant links.

On-premises

- Setup database admin user in SnapCenter
- SnapCenter plugin installation prerequisites
- SnapCenter host plugin installation
- DB resource discovery
- Setup storage cluster peering and DB volume replication
- Add CVO database storage SVM to SnapCenter
• Setup database backup policy in SnapCenter
• Implement backup policy to protect database
• Validate backup

AWS public cloud

• Pre-flight check
• Steps to deploy Cloud Manager and Cloud Volumes ONTAP in AWS
• Deploy EC2 compute instance for database workload

Click the following links for details:

On Premises, Public Cloud - AWS

Getting started on premises

The NetApp SnapCenter tool uses role based access control (RBAC) to manage user resources access and permission grants, and SnapCenter installation creates prepopulated roles. You can also create custom roles based on your needs or applications.

On Premises

1. Setup database admin user in SnapCenter

It makes sense to have a dedicated admin user ID for each database platform supported by SnapCenter for database backup, restoration, and/or disaster recovery. You can also use a single ID to manage all databases. In our test cases and demonstration, we created a dedicated admin user for both Oracle and SQL Server, respectively.

Certain SnapCenter resources can only be provisioned with the SnapCenterAdmin role. Resources can then be assigned to other user IDs for access.

In a pre-installed and configured on-premises SnapCenter environment, the following tasks might have already been completed. If not, the following steps create a database admin user:

1. Add the admin user to Windows Active Directory.
2. Log into SnapCenter using an ID granted with the SnapCenterAdmin role.
3. Navigate to the Access tab under Settings and Users, and click Add to add a new user. The new user ID is linked to the admin user created in Windows Active Directory in step 1. Assign the proper role to the user as needed. Assign resources to the admin user as applicable.
2. SnapCenter plugin installation prerequisites

SnapCenter performs backup, restore, clone, and other functions by using a plugin agent running on the DB hosts. It connects to the database host and database via credentials configured under the Setting and Credentials tab for plugin installation and other management functions. There are specific privilege requirements based on the target host type, such as Linux or Windows, as well as the type of database.

DB hosts credentials must be configured before SnapCenter plugin installation. Generally, you want to use an administrator user accounts on the DB host as your host connection credentials for plugin installation. You can also grant the same user ID for database access using OS-based authentication. On the other hand, you can also employ database authentication with different database user IDs for DB management access. If you decide to use OS-based authentication, the OS admin user ID must be granted DB access. For Windows domain-based SQL Server installation, a domain admin account can be used to manage all SQL Servers within the domain.

Windows host for SQL server:

1. If you are using Windows credentials for authentication, you must set up your credential before installing plugins.
2. If you are using a SQL Server instance for authentication, you must add the credentials after installing plugins.
3. If you have enabled SQL authentication while setting up the credentials, the discovered instance or database is shown with a red lock icon. If the lock icon appears, you must specify the instance or database credentials to successfully add the instance or database to a resource group.
4. You must assign the credential to a RBAC user without sysadmin access when the following conditions are met:
   - The credential is assigned to a SQL instance.
   - The SQL instance or host is assigned to an RBAC user.
   - The RBAC DB admin user must have both the resource group and backup privileges.

Unix host for Oracle:

1. You must have enabled the password-based SSH connection for the root or non-root user by editing ssdh.conf and restarting the ssdh service. Password-based SSH authentication on AWS instance is turned off by default.
2. Configure the sudo privileges for the non-root user to install and start the plugin process. After installing the plugin, the processes run as an effective root user.
3. Create credentials with the Linux authentication mode for the install user.
4. You must install Java 1.8.x (64-bit) on your Linux host.
5. Installation of the Oracle database plugin also installs the SnapCenter plugin for Unix.

3. SnapCenter host plugin installation

   Before attempting to install SnapCenter plugins on cloud DB server instances, make sure that all configuration steps have been completed as listed in the relevant cloud section for compute instance deployment.

The following steps illustrate how a database host is added to SnapCenter while a SnapCenter plugin is installed on the host. The procedure applies to adding both on-premises hosts and cloud hosts. The following demonstration adds a Windows or a Linux host residing in AWS.

Configure SnapCenter VMware global settings

Navigate to Settings > Global Settings. Select "VMs have iSCSI direct attached disks or NFS for all the hosts" under Hypervisor Settings and click Update.

Add Windows host and installation of plugin on the host

1. Log into SnapCenter with a user ID with SnapCenterAdmin privileges.
2. Click the Hosts tab from the left-hand menu, and then click Add to open the Add Host workflow.
3. Choose Windows for Host Type; the Host Name can be either a host name or an IP address. The host name must be resolved to the correct host IP address from the SnapCenter host. Choose the host credentials created in step 2. Choose Microsoft Windows and Microsoft SQL Server as the plugin packages to be installed.
4. After the plugin is installed on a Windows host, its Overall Status is shown as "Configure log directory."
5. Click the Host Name to open the SQL Server log directory configuration.

6. Click "Configure log directory" to open "Configure Plug-in for SQL Server."

7. Click Browse to discover NetApp storage so that a log directory can be set; SnapCenter uses this log directory to roll up the SQL server transaction log files. Then click Save.
For NetApp storage provisioned to a DB host to be discovered, the storage (on-prem or CVO) must be added to SnapCenter, as illustrated in step 6 for CVO as an example.

8. After the log directory is configured, the Windows host plugin Overall Status is changed to Running.

9. To assign the host to the database management user ID, navigate to the Access tab under Settings and Users, click the database management user ID (in our case the sqldba that the host needs to be assigned to), and click Save to complete host resource assignment.
Add Unix host and installation of plugin on the host

1. Log into SnapCenter with a user ID with SnapCenterAdmin privileges.

2. Click the Hosts tab from left-hand menu, and click Add to open the Add Host workflow.

3. Choose Linux as the Host Type. The Host Name can be either the host name or an IP address. However, the host name must be resolved to correct host IP address from SnapCenter host. Choose host credentials created in step 2. The host credentials require sudo privileges. Check Oracle Database as the plug-in to be installed, which installs both Oracle and Linux host plugins.

4. Click More Options and select "Skip preinstall checks." You are prompted to confirm the skipping of the preinstall check. Click Yes and then Save.
5. Click Submit to start the plugin installation. You are prompted to Confirm Fingerprint as shown below.

6. SnapCenter performs host validation and registration, and then the plugin is installed on the Linux host. The status is changed from Installing Plugin to Running.

7. Assign the newly added host to the proper database management user ID (in our case, oradba).
4. Database resource discovery

With successful plugin installation, the database resources on the host can be immediately discovered. Click the Resources tab in the left-hand menu. Depending on the type of database platform, a number of views are available, such as the database, resources group, and so on. You might need to click the Refresh Resources tab if the resources on the host are not discovered and displayed.
When the database is initially discovered, the Overall Status is shown as "Not protected." The previous screenshot shows an Oracle database not protected yet by a backup policy.

When a backup configuration or policy is set up and a backup has been executed, the Overall Status for the database shows the backup status as "Backup succeeded" and the timestamp of the last backup. The following screenshot shows the backup status of a SQL Server user database.

If database access credentials are not properly set up, a red lock button indicates that the database is not accessible. For example, if Windows credentials do not have sysadmin access to a database instance, then database credentials must be reconfigured to unlock the red lock.

After the appropriate credentials are configured either at the Windows level or the database level, the red lock disappears and SQL Server Type information is gathered and reviewed.
5. Setup storage cluster peering and DB volumes replication

To protect your on-premises database data using a public cloud as the target destination, on-premises ONTAP cluster database volumes are replicated to the cloud CVO using NetApp SnapMirror technology. The replicated target volumes can then be cloned for DEV/OPS or disaster recovery. The following high-level steps enable you to set up cluster peering and DB volumes replication.

1. Configure intercluster LIFs for cluster peering on both the on-premises cluster and the CVO cluster instance. This step can be performed with ONTAP System Manager. A default CVO deployment has inter-cluster LIFs configured automatically.

   On-premises cluster:

   ![On-premises cluster image]

   Target CVO cluster:

   ![Target CVO cluster image]

2. With the intercluster LIFs configured, cluster peering and volume replication can be set up by using drag-and-drop in NetApp Cloud Manager. See "Getting Started - AWS Public Cloud" for details.

   Alternatively, cluster peering and DB volume replication can be performed by using ONTAP System Manager as follows:

3. Log into ONTAP System Manager. Navigate to Cluster > Settings and click Peer Cluster to set up cluster peering with the CVO instance in the cloud.
4. Go to the Volumes tab. Select the database volume to be replicated and click Protect.

5. Set the protection policy to Asynchronous. Select the destination cluster and storage SVM.
6. Validate that the volume is synced between the source and target and that the replication relationship is healthy.

6. Add CVO database storage SVM to SnapCenter

1. Log into SnapCenter with a user ID with SnapCenterAdmin privileges.

2. Click the Storage System tab from the menu, and then click New to add a CVO storage SVM that hosts replicated target database volumes to SnapCenter. Enter the cluster management IP in the Storage System field, and enter the appropriate username and password.
3. Click More Options to open additional storage configuration options. In the Platform field, select Cloud Volumes ONTAP, check Secondary, and then click Save.

4. Assign the storage systems to SnapCenter database management user IDs as shown in 3. SnapCenter host plugin installation.

7. Setup database backup policy in SnapCenter

The following procedures demonstrates how to create a full database or log file backup policy. The policy can then be implemented to protect databases resources. The recovery point objective (RPO) or recovery time objective (RTO) dictates the frequency of database and/or log backups.
Create a full database backup policy for Oracle

1. Log into SnapCenter as a database management user ID, click Settings, and then click Policies.

2. Click New to launch a new backup policy creation workflow or choose an existing policy for modification.

3. Select the backup type and schedule frequency.
4. Set the backup retention setting. This defines how many full database backup copies to keep.
5. Select the secondary replication options to push local primary snapshots backups to be replicated to a secondary location in cloud.
6. Specify any optional script to run before and after a backup run.
7. Run backup verification if desired.
8. Summary.
Create a database log backup policy for Oracle

1. Log into SnapCenter with a database management user ID, click Settings, and then click Policies.
2. Click New to launch a new backup policy creation workflow, or choose an existing policy for modification.
3. Select the backup type and schedule frequency.
4. Set the log retention period.
5. Enable replication to a secondary location in the public cloud.
6. Specify any optional scripts to run before and after log backup.
7. Specify any backup verification scripts.
8. Summary.
Create a full database backup policy for SQL

1. Log into SnapCenter with a database management user ID, click Settings, and then click Policies.

2. Click New to launch a new backup policy creation workflow, or choose an existing policy for modification.
3. Define the backup option and schedule frequency. For SQL Server configured with an availability group, a preferred backup replica can be set.
4. Set the backup retention period.
5. Enable backup copy replication to a secondary location in cloud.
6. Specify any optional scripts to run before or after a backup job.
7. Specify the options to run backup verification.
8. Summary.
Create a database log backup policy for SQL.

1. Log into SnapCenter with a database management user ID, click Settings > Policies, and then New to launch a new policy creation workflow.
2. Define the log backup option and schedule frequency. For SQL Server configured with an availability group, a preferred backup replica can be set.
3. SQL server data backup policy defines the log backup retention; accept the defaults here.
4. Enable log backup replication to secondary in the cloud.
5. Specify any optional scripts to run before or after a backup job.
8. Implement backup policy to protect database

SnapCenter uses a resource group to backup a database in a logical grouping of database resources, such as multiple databases hosted on a server, a database sharing the same storage volumes, multiple databases supporting a business application, and so on. Protecting a single database creates a resource group of its own. The following procedures demonstrate how to implement a backup policy created in section 7 to protect Oracle and SQL Server databases.

Create a resource group for full backup of Oracle

1. Log into SnapCenter with a database management user ID, and navigate to the Resources tab. In the View drop-down list, choose either Database or Resource Group to launch the resource group creation workflow.

2. Provide a name and tags for the resource group. You can define a naming format for the Snapshot copy and bypass the redundant archive log destination if configured.
3. Add database resources to the resource group.

4. Select a full backup policy created in section 7 from the drop-down list.

5. Click the (+) sign to configure the desired backup schedule.
6. Click Load Locators to load the source and destination volume.

7. Configure the SMTP server for email notification if desired.
8. Summary.

Create a resource group for log backup of Oracle

1. Log into SnapCenter with a database management user ID, and navigate to the Resources tab. In the View drop-down list, choose either Database or Resource Group to launch the resource group creation workflow.

2. Provide a name and tags for the resource group. You can define a naming format for the Snapshot copy and bypass the redundant archive log destination if configured.
3. Add database resources to the resource group.

4. Select a log backup policy created in section 7 from the drop-down list.

5. Click on the (+) sign to configure the desired backup schedule.
6. If backup verification is configured, it displays here.

7. Configure an SMTP server for email notification if desired.
Create a resource group for full backup of SQL Server

1. Log into SnapCenter with a database management user ID, and navigate to the Resources tab. In the View drop-down list, choose either a Database or Resource Group to launch the resource group creation workflow. Provide a name and tags for the resource group. You can define a naming format for the Snapshot copy.
2. Select the database resources to be backed up.

3. Select a full SQL backup policy created in section 7.
4. Add exact timing for backups as well as the frequency.

5. Choose the verification server for the backup on secondary if backup verification is to be performed. Click Load Locator to populate the secondary storage location.

6. Configure the SMTP server for email notification if desired.
7. Summary.

Create a resource group for log backup of SQL Server

1. Log into SnapCenter with a database management user ID, and navigate to the Resources tab. In the View drop-down list, choose either a Database or Resource Group to launch the resource group creation workflow. Provide the name and tags for the resource group. You can define a naming format for the Snapshot copy.
2. Select the database resources to be backed up.

3. Select a SQL log backup policy created in section 7.
4. Add exact timing for the backup as well as the frequency.

5. Choose the verification server for the backup on secondary if backup verification is to be performed. Click the Load Locator to populate the secondary storage location.

6. Configure the SMTP server for email notification if desired.
7. Summary.

9. Validate backup

After database backup resource groups are created to protect database resources, the backup jobs run according to the predefined schedule. Check the job execution status under the Monitor tab.

Go to the Resources tab, click the database name to view details of database backup, and toggle between Local copies and mirror copies to verify that Snapshot backups are replicated to a secondary location in the
At this point, database backup copies in the cloud are ready to clone to run dev/test processes or for disaster recovery in the event of a primary failure.

**Getting Started with AWS public cloud**

This section describes the process of deploying Cloud Manager and Cloud Volumes ONTAP in AWS.

**AWS public cloud**

To make things easier to follow, we have created this document based on a deployment in AWS. However, the process is very similar for Azure and GCP.

**1. Pre-flight check**

Before deployment, make sure that the infrastructure is in place to allow for the deployment in the next stage. This includes the following:

- AWS account
- VPC in your region of choice
- Subnet with access to the public internet
- Permissions to add IAM roles into your AWS account
- A secret key and access key for your AWS user

**2. Steps to deploy Cloud Manager and Cloud Volumes ONTAP in AWS**

There are many methods for deploying Cloud Manager and Cloud Volumes ONTAP; this method is the simplest but requires the most permissions. If this method is not appropriate for your AWS environment, please consult the NetApp Cloud Documentation.

**Deploy the Cloud Manager connector**

1. Navigate to NetApp Cloud Central and log in or sign up.
2. After you log in, you should be taken to the Canvas.
3. Click "Add Working Environment" and choose Cloud Volumes ONTAP in AWS. Here, you also choose whether you want to deploy a single node system or a high availability pair. I have chosen to deploy a high availability pair.

4. If no connector has been created, a pop-up appears asking you to create a connector.
5. Click Let's Start, and then choose AWS.

6. Enter your secret key and access key. Make sure that your user has the correct permissions outlined on the NetApp policies page.
7. Give the connector a name and either use a predefined role as described on the NetApp policies page or ask Cloud Manager to create the role for you.

8. Give the networking information needed to deploy the connector. Verify that outbound internet access is enabled by:
   a. Giving the connector a public IP address
   b. Giving the connector a proxy to work through
   c. Giving the connector a route to the public internet through an Internet Gateway
9. Provide communication with the connector via SSH, HTTP, and HTTPs by either providing a security group or creating a new security group. I have enabled access to the connector from my IP address only.

10. Review the information on the summary page and click Add to deploy the connector.
11. The connector now deploys using a cloud formation stack. You can monitor its progress from Cloud Manager or through AWS.

12. When the deployment is complete, a success page appears.

---

**Deploy Cloud Volumes ONTAP**

1. Select AWS and the type of deployment based on your requirements.
2. If no subscription has been assigned and you wish to purchase with PAYGO, choose Edit Credentials.

3. Choose Add Subscription.
4. Choose the type of contract that you wish to subscribe to. I chose Pay-as-you-go.

5. You are redirected to AWS; choose Continue to Subscribe.

6. Subscribe and you are redirected back to NetApp Cloud Central. If you have already subscribed and don’t get redirected, choose the "Click here" link.
7. You are redirected to Cloud Central where you must name your subscription and assign it to your Cloud Central account.

8. When successful, a check mark page appears. Navigate back to your Cloud Manager tab.
9. The subscription now appears in Cloud Central. Click Apply to continue.

10. Enter the working environment details such as:
   a. Cluster name
   b. Cluster password
   c. AWS tags (Optional)
11. Choose which additional services you would like to deploy. To discover more about these services, visit the NetApp Cloud Homepage.

12. Choose whether to deploy in multiple availability zones (requires three subnets, each in a different AZ), or a single availability zone. I chose multiple AZs.
13. Choose the region, VPC, and security group for the cluster to be deployed into. In this section, you also assign the availability zones per node (and mediator) as well as the subnets that they occupy.

14. Choose the connection methods for the nodes as well as the mediator.
The mediator requires communication with the AWS APIs. A public IP address is not required so long as the APIs are reachable after the mediator EC2 instance has been deployed.

1. Floating IP addresses are used to allow access to the various IP addresses that Cloud Volumes ONTAP uses, including cluster management and data serving IPs. These must be addresses that are not already routable within your network and are added to route tables in your AWS environment. These are required to enable consistent IP addresses for an HA pair during failover. More information about floating IP addresses can be found in the NetApp Cloud Documentation.

2. Select which route tables the floating IP addresses are added to. These route tables are used by clients to communicate with Cloud Volumes ONTAP.

3. Choose whether to enable AWS managed encryption or AWS KMS to encrypt the ONTAP root, boot, and data disks.
4. Choose your licensing model. If you don’t know which to choose, contact your NetApp representative.

5. Select which configuration best suits your use case. This is related to the sizing considerations covered in the prerequisites page.
6. Optionally, create a volume. This is not required, because the next steps use SnapMirror, which creates the volumes for us.

7. Review the selections made and tick the boxes to verify that you understand that Cloud Manager deploys resources into your AWS environment. When ready, click Go.

8. Cloud Volumes ONTAP now starts its deployment process. Cloud Manager uses AWS APIs and cloud formation stacks to deploy Cloud Volumes ONTAP. It then configures the system to your specifications, giving you a ready-to-go system that can be instantly utilized. The timing for this process varies depending on the selections made.
9. You can monitor the progress by navigating to the Timeline.

10. The Timeline acts as an audit of all actions performed in Cloud Manager. You can view all of the API calls that are made by Cloud Manager during setup to both AWS as well as the ONTAP cluster. This can also be effectively used to troubleshoot any issues that you face.
11. After deployment is complete, the CVO cluster appears on the Canvas, which the current capacity. The ONTAP cluster in its current state is fully configured to allow a true, out-of-the-box experience.

Configure SnapMirror from on-premises to cloud

Now that you have a source ONTAP system and a destination ONTAP system deployed, you can replicate volumes containing database data into the cloud.

For a guide on compatible ONTAP versions for SnapMirror, see the SnapMirror Compatibility Matrix.

1. Click the source ONTAP system (on-premises) and either drag and drop it to the destination, select Replication > Enable, or select Replication > Menu > Replicate.
Select Enable.

SERVICES

Replication
- Off

Enable

Or Options.
2. If you did not drag and drop, choose the destination cluster to replicate to.
3. Choose the volume that you’d like to replicate. We replicated the data and all log volumes.

4. Choose the destination disk type and tiering policy. For disaster recovery, we recommend an SSD as the disk type and to maintain data tiering. Data tiering tiers the mirrored data into low-cost object storage and saves you money on local disks. When you break the relationship or clone the volume, the data uses the fast, local storage.
5. Select the destination volume name: we chose `[source_volume_name]_dr`.

6. Select the maximum transfer rate for the replication. This enables you to save bandwidth if you have a low bandwidth connection to the cloud such as a VPN.
Max Transfer Rate

You should limit the transfer rate. An unlimited rate might negatively impact the performance of other applications and it might impact your Internet performance.

- Limited to: 100 MB/s
- Unlimited (recommended for DR only machines)

7. Define the replication policy. We chose a Mirror, which takes the most recent dataset and replicates that into the destination volume. You could also choose a different policy based on your requirements.

**Replication Policy**

<table>
<thead>
<tr>
<th>Default Policies</th>
<th>Additional Policies</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Mirror" /></td>
<td><img src="image" alt="Mirror and Backup (1 month retention)" /></td>
</tr>
<tr>
<td>Typically used for disaster recovery</td>
<td>Configures disaster recovery and long-term retention of backups on the same destination volume</td>
</tr>
</tbody>
</table>

8. Choose the schedule for triggering replication. NetApp recommends setting a "daily" schedule of for the data volume and an "hourly" schedule for the log volumes, although this can be changed based on requirements.
9. Review the information entered, click Go to trigger the cluster peer and SVM peer (if this is your first time replicating between the two clusters), and then implement and initialize the SnapMirror relationship.

10. Continue this process for data volumes and log volumes.

11. To check all of your relationships, navigate to the Replication tab inside Cloud Manager. Here you can manage your relationships and check on their status.

12. After all the volumes have been replicated, you are in a steady state and ready to move on to the disaster recovery and dev/test workflows.
3. Deploy EC2 compute instance for database workload

AWS has preconfigured EC2 compute instances for various workloads. The choice of instance type determines the number of CPU cores, memory capacity, storage type and capacity, and network performance. For the use cases, with the exception of the OS partition, the main storage to run database workload is allocated from CVO or the FSx ONTAP storage engine. Therefore, the main factors to consider are the choice of CPU cores, memory, and network performance level. Typical AWS EC2 instance types can be found here: EC2 Instance Type.

Sizing the compute instance

1. Select the right instance type based on the required workload. Factors to consider include the number of business transactions to be supported, the number of concurrent users, data set sizing, and so on.
2. EC2 instance deployment can be launched through the EC2 Dashboard. The exact deployment procedures are beyond the scope of this solution. See Amazon EC2 for details.

Linux instance configuration for Oracle workload

This section contain additional configuration steps after an EC2 Linux instance is deployed.

1. Add an Oracle standby instance to the DNS server for name resolution within the SnapCenter management domain.
2. Add a Linux management user ID as the SnapCenter OS credentials with sudo permissions without a password. Enable the ID with SSH password authentication on the EC2 instance. (By default, SSH password authentication and passwordless sudo is turned off on EC2 instances.)
3. Configure Oracle installation to match with on-premises Oracle installation such as OS patches, Oracle versions and patches, and so on.
4. NetApp Ansible DB automation roles can be leveraged to configure EC2 instances for database dev/test and disaster recovery use cases. The automation code can be download from the NetApp public GitHub site: Oracle 19c Automated Deployment. The goal is to install and configure a database software stack on an EC2 instance to match on-premises OS and database configurations.

Windows instance configuration for SQL Server workload

This section lists additional configuration steps after an EC2 Windows instance is initially deployed.

1. Retrieve the Windows administrator password to log in to an instance via RDP.
2. Disable the Windows firewall, join the host to Windows SnapCenter domain, and add the instance to the DNS server for name resolution.
3. Provision a SnapCenter log volume to store SQL Server log files.
4. Configure iSCSI on the Windows host to mount the volume and format the disk drive.
5. Again, many of the previous tasks can be automated with the NetApp automation solution for SQL Server. Check the NetApp automation public GitHub site for newly published roles and solutions: NetApp Automation.

Workflow for dev/test bursting to cloud

The agility of the public cloud, the time to value, and the cost savings are all meaningful value propositions for enterprises adopting the public cloud for database application development and testing effort. There is no better tool than SnapCenter to make this a
SnapCenter can not only protect your production database on-premises, but can also quickly clone a copy for application development or code testing in the public cloud while consuming very little extra storage. Following are details of the step-by-step processes for using this tool.

**Clone an Oracle Database for dev/test from a replicated snapshot backup**

1. Log into SnapCenter with a database management user ID for Oracle. Navigate to the Resources tab, which shows the Oracle databases being protected by SnapCenter.

2. Click the intended on-premises database name for the backup topology and the detailed view. If a secondary replicated location is enabled, it shows linked mirror backups.

3. Toggled to the mirrored backups view by clicking mirrored backups. The secondary mirror backup(s) is then displayed.
4. Choose a mirrored secondary database backup copy to be cloned and determine a recovery point either by time and system change number or by SCN. Generally, the recovery point should be trailing the full database backup time or SCN to be cloned. After a recovery point is decided, the required log file backup must be mounted for recovery. The log file backup should be mounted to target DB server where the clone database is to be hosted.

Mount backups

Choose the host to mount the backup: ore-standby.demo.netapp.com

Mount path: /var/opt/snapcenter/scv/backup_mount/rhel2_cdb2_09-17-2021_14:35:01.4997_1/cdb2

Secondary storage location: Snap Vault / Snap Mirror

<table>
<thead>
<tr>
<th>Source Volume</th>
<th>Destination Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>svm_onlrem/rhel2_u03</td>
<td>svv_hybridcvorhel2_u03_d</td>
</tr>
</tbody>
</table>

[Mount] [Cancel]
If log pruning is enabled and the recovery point is extended beyond the last log pruning, multiple archive log backups might need to be mounted.

5. Highlight the full database backup copy to be cloned, and then click the clone button to start the DB clone Workflow.

6. Choose a proper clone DB SID for a complete container database or CDB clone.
7. Select the target clone host in the cloud, and datafile, control file, and redo log directories are created by the clone workflow.
8. The None credential name is used for OS-based authentication, which renders the database port irrelevant. Fill in the proper Oracle Home, Oracle OS User, and Oracle OS Group as configured in the target clone DB server.
9. Specify the scripts to run before clone operation. More importantly, the database instance parameter can be adjusted or defined here.
10. Specify the recovery point either by the date and time or SCN. Until Cancel recovers the database up to the available archive logs. Specify the external archive log location from the target host where the archive log volume is mounted. If target server Oracle owner is different from the on-premises production server, verify that the archive log directory is readable by the target server Oracle owner.
11. Configure the SMTP server for email notification if desired.
12. Clone summary.
13. You should validate after cloning to make sure that the cloned database is operational. Some additional tasks, such as starting up the listener or turning off the DB log archive mode, can be performed on the dev/test database.
Clone a SQL database for dev/test from a replicated Snapshot backup

1. Log into SnapCenter with a database management user ID for SQL Server. Navigate to the Resources tab, which shows the SQL Server user databases being protected by SnapCenter and a target standby SQL instance in the public cloud.

2. Click on the intended on-premises SQL Server user database name for the backups topology and detailed view. If a secondary replicated location is enabled, it shows linked mirror backups.

3. Toggle to the Mirrored Backups view by clicking Mirrored Backups. Secondary Mirror Backup(s) are then displayed. Because SnapCenter backs up the SQL Server transaction log to a dedicated drive for recovery, only full database backups are displayed here.
4. Choose a backup copy, and then click the Clone button to launch the Clone from Backup workflow.

5. Select a cloud server as the target clone server, clone instance name, and clone database name. Choose either an auto-assign mount point or a user-defined mount point path.
6. Determine a recovery point either by a log backup time or by a specific date and time.
7. Specify optional scripts to run before and after the cloning operation.
8. Configure an SMTP server if email notification is desired.
9. Clone Summary.
Monitor the job status and validate that the intended user database has been attached to a target SQL instance in the cloud clone server.

Post-clone configuration

1. An Oracle production database on-premises is usually running in log archive mode. This mode is not necessary for a development or test database. To turn off log archive mode, log into the Oracle DB as sysdba, execute a log mode change command, and start the database for access.

2. Configure an Oracle listener, or register the newly cloned DB with an existing listener for user access.

3. For SQL Server, change the log mode from Full to Easy so that the SQL Server dev/test log file can be readily shrunk when it is filling up the log volume.
Refresh clone database

1. Drop cloned databases and clean up the cloud DB server environment. Then follow the previous procedures to clone a new DB with fresh data. It only takes few minutes to clone a new database.

2. Shutdown the clone database, run a clone refresh command by using the CLI. See the following SnapCenter documentation for details: Refresh a clone.

Where to go for help?

If you need help with this solution and use cases, join the NetApp Solution Automation community support Slack channel and look for the solution-automation channel to post your questions or inquires.

Disaster recovery workflow

Enterprises have embraced the public cloud as a viable resource and destination for disaster recovery. SnapCenter makes this process as seamless as possible. This disaster recovery workflow is very similar to the clone workflow, but database recovery runs through the last available log that was replicated to cloud to recover all the business transactions possible. However, there are additional pre-configuration and post-configuration steps specific to disaster recovery.

Clone an on-premises Oracle production DB to cloud for DR

1. To validate that the clone recovery runs through last available log, we created a small test table and inserted a row. The test data would be recovered after a full recovery to last available log.

2. Log into SnapCenter as a database management user ID for Oracle. Navigate to the Resources tab, which shows the Oracle databases being protected by SnapCenter.
3. Select the Oracle log resource group and click Backup Now to manually run an Oracle log backup to flush the latest transaction to the destination in the cloud. In a real DR scenario, the last transaction recoverable depends on the database log volume replication frequency to the cloud, which in turn depends on the RTO or RPO policy of the company.

Asynchronous SnapMirror loses data that has not made it to the cloud destination in the database log backup interval in a disaster recovery scenario. To minimize data loss, more frequent log backup can be scheduled. However there is a limit to the log backup frequency that is technically achievable.

4. Select the last log backup on the Secondary Mirror Backup(s), and mount the log backup.
5. Select the last full database backup and click Clone to initiate the clone workflow.
6. Select a unique clone DB ID on the host.

7. Provision a log volume and mount it to the target DR server for the Oracle flash recovery area and online logs.
The Oracle clone procedure does not create a log volume, which needs to be provisioned on the DR server before cloning.

8. Select the target clone host and location to place the data files, control files, and redo logs.

9. Select the credentials for the clone. Fill in the details of the Oracle home configuration on the target server.
10. Specify the scripts to run before cloning. Database parameters can be adjusted if needed.
11. Select Until Cancel as the recovery option so that the recovery runs through all available archive logs to recoup the last transaction replicated to the secondary cloud location.
12. Configure the SMTP server for email notification if needed.
13. DR clone summary.
14. Cloned DBs are registered with SnapCenter immediately after clone completion and are then available for backup protection.

Post DR clone validation and configuration for Oracle

1. Validate the last test transaction that has been flushed, replicated, and recovered at the DR location in the cloud.
2. Configure the flash recovery area.

3. Configure the Oracle listener for user access.

4. Split the cloned volume off of the replicated source volume.

5. Reverse replication from the cloud to on-premises and rebuild the failed on-premises database server.

   Clone split may incur temporary storage space utilization that is much higher than normal operation. However, after the on-premises DB server is rebuilt, extra space can be released.

**Clone an on-premises SQL production DB to cloud for DR**

1. Similarly, to validate that the SQL clone recovery ran through last available log, we created a small test table and inserted a row. The test data would be recovered after a full recovery to the last available log.
2. Log into SnapCenter with a database management user ID for SQL Server. Navigate to the Resources tab, which shows the SQL Server protection resources group.

3. Manually run a log backup to flush the last transaction to be replicated to secondary storage in the public cloud.

4. Select the last full SQL Server backup for the clone.
5. Set the clone setting such as the Clone Server, Clone Instance, Clone Name, and mount option. The secondary storage location where cloning is performed is auto-populated.

6. Select all log backups to be applied.
7. Specify any optional scripts to run before or after cloning.
8. Specify an SMTP server if email notification is desired.
9. DR clone summary. Cloned databases are immediately registered with SnapCenter and available for backup protection.
Post DR clone validation and configuration for SQL

1. Monitor clone job status.

2. Validate that last transaction has been replicated and recovered with all log file clones and recovery.
3. Configure a new SnapCenter log directory on the DR server for SQL Server log backup.

4. Split the cloned volume off of the replicated source volume.

5. Reverse replication from the cloud to on-premises and rebuild the failed on-premises database server.

Where to go for help?

If you need help with this solution and use cases, please join the NetApp Solution Automation community support Slack channel and look for the solution-automation channel to post your questions or inquires.

DB Automation Toolkits

Automated Oracle Migration

NetApp Solutions Engineering Team

Purpose

This toolkit automates Oracle database migration from on-premises to AWS cloud with FSx ONTAP storage and EC2 compute instance as target infrastructure. It assumes the customer already has an on-premises Oracle database deployed in the CDB/PDB model. The toolkit will allow the customer to relocate a named PDB from a container database on an Oracle host using the Oracle PDB relocation procedure with a maximum availability option. That means the source PDB on any on-premises storage array relocates to a new container database with minimal service interruption. The Oracle relocation procedure will move the Oracle data files while database is online. It subsequently reroutes user sessions from on-premises to the relocated database services at the time of switching over when all data files move over to AWS cloud. The underlined technology is proven Oracle PDB hot clone methodology.

Although the migration toolkit is developed and validated on AWS cloud infrastructure, it builds on Oracle application-level solutions. Therefore, the toolkit is applicable to other public cloud platforms, such as Azure, GCP, etc.

This solution addresses the following use cases:

• Create migration user and grant required privileges at on-prem source DB server.

• Relocate a PDB from on-premises CDB to a target CDB in cloud while the source PDB is online until switch over.
Audience

This solution is intended for the following people:

• A DBA who migrates Oracle databases from on-premises to AWS cloud.
• A database solution architect who is interested in Oracle database migration from on-premises to AWS cloud.
• A storage administrator who manages AWS FSx ONTAP storage that supports Oracle databases.
• An application owner who likes to migrate Oracle database from on-premises to AWS cloud.

License

By accessing, downloading, installing or using the content in this GitHub repository, you agree the terms of the License laid out in License file.

There are certain restrictions around producing and/or sharing any derivative works with the content in this GitHub repository. Please make sure you read the terms of the License before using the content. If you do not agree to all of the terms, do not access, download or use the content in this repository.

Solution deployment

Prerequisites for deployment

Deployment requires the following prerequisites.

- Ansible v.2.10 and higher
- ONTAP collection 21.19.1
- Python 3
- Python libraries:
  - netapp-lib
  - xmltodict
  - jmespath

- Source Oracle CDB with PDBs on-premises
- Target Oracle CDB in AWS hosted on FSx and EC2 instance
- Source and target CDB on same version and with same options installed

- Network connectivity
  - Ansible controller to source CDB
  - Ansible controller to target CDB
  - Source CDB to target CDB on Oracle listener port (typical 1521)
Host variables configuration

Host variables are defined in host_vars directory named as {{ host_name }}.yml. An example host variable file host_name.yml is included to demonstrate typical configuration. Following are key considerations:

Source Oracle CDB - define host specific variables for the on-prem CDB
- ansible_host: IP address of source database server host
- source_oracle_sid: source Oracle CDB instance ID
- source_pdb_name: source PDB name to migrate to cloud
- source_file_directory: file directory of source PDB data files
- target_file_directory: file directory of migrated PDB data files

Target Oracle CDB - define host specific variables for the target CDB including some variables for on-prem CDB
- ansible_host: IP address of target database server host
- target_oracle_sid: target Oracle CDB instance ID
- target_pdb_name: target PDB name to be migrated to cloud (for max availability option, the source and target PDB name must be the same)
- source_oracle_sid: source Oracle CDB instance ID
- source_pdb_name: source PDB name to be migrated to cloud
- source_port: source Oracle CDB listener port
- source_oracle_domain: source Oracle database domain name
- source_file_directory: file directory of source PDB data files
- target_file_directory: file directory of migrated PDB data files

DB server host file configuration
AWS EC2 instance use IP address for host naming by default. If you use different name in hosts file for Ansible, setup host naming resolution in /etc/hosts file for both source and target server. Following is an example.

```
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
172.30.15.96 source_db_server
172.30.15.107 target_db_server
```

Playbook execution - executed in sequence

1. Install Ansible controller prerequisites.

   `ansible-playbook -i hosts requirements.yml`

   `ansible-galaxy collection install -r collections/requirements.yml --force`

2. Execute pre-migration tasks against on-prem server - assuming admin is ssh user for connection to on-prem Oracle host with sudo permission.

   `ansible-playbook -i hosts ora_pdb_relocate.yml -u admin -k -K -t ora_pdb_relo_onprem`

3. Execute Oracle PDB relocation from on-prem CDB to target CDB in AWS EC2 instance - assuming ec2-user for EC2 DB instance connection, and db1.pem with ec2-user ssh key pairs.

   `ansible-playbook -i hosts ora_pdb_relocate.yml -u ec2-user --private -key db1.pem -t ora_pdb_relo_primary`

Where to find additional information

To learn more about the NetApp solution automation, review the following website NetApp Solution Automation

Automated Oracle HA/DR in AWS FSx ONTAP

NetApp Solutions Engineering Team
Purpose

This toolkit automates the tasks of setting up and managing a High Availability and Disaster Recovery (HR/DR) environment for Oracle database deployed in AWS cloud with FSx for ONTAP storage and EC2 compute instances.

This solution addresses the following use cases:

- Setup HA/DR target host - kernel configuration, Oracle configuration to match up with source server host.
- Setup FSx ONTAP - cluster peering, vserver peering, Oracle volumes snapmirror relationship setup from source to target.
- Backup Oracle database data via snapshot - execute off crontab
- Backup Oracle database archive log via snapshot - execute off crontab
- Run failover and recovery on HA/DR host - test and validate HA/DR environment
- Run resync after failover test - re-establish database volumes snapmirror relationship in HA/DR mode

Audience

This solution is intended for the following people:

- A DBA who set up Oracle database in AWS for high availability, data protection, and disaster recovery.
- A database solution architect who is interested in storage level Oracle HA/DR solution in the AWS cloud.
- A storage administrator who manages AWS FSx ONTAP storage that supports Oracle databases.
- An application owner who like to stand up Oracle database for HA/DR in AWS FSx/EC2 environment.

License

By accessing, downloading, installing or using the content in this GitHub repository, you agree the terms of the License laid out in License file.

There are certain restrictions around producing and/or sharing any derivative works with the content in this GitHub repository. Please make sure you read the terms of the License before using the content. If you do not agree to all of the terms, do not access, download or use the content in this repository.

Solution deployment

Prerequisites for deployment
Deployment requires the following prerequisites.

<table>
<thead>
<tr>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ansible v.2.10 and higher</td>
</tr>
<tr>
<td>ONTAP collection 21.19.1</td>
</tr>
<tr>
<td>Python 3</td>
</tr>
<tr>
<td>Python libraries:</td>
</tr>
<tr>
<td>netapp-lib</td>
</tr>
<tr>
<td>xmltodict</td>
</tr>
<tr>
<td>jmespath</td>
</tr>
</tbody>
</table>

AWS FSx storage as is available

AWS EC2 Instance
- RHEL 7/8, Oracle Linux 7/8
- Network interfaces for NFS, public (internet) and optional management
- Existing Oracle environment on source, and the equivalent Linux operating system at the target

Download the toolkit

```bash
git clone https://github.com/NetApp/na_ora_hadr_failover_resync.git
```

Global variables configuration
The Ansible playbooks are variable driven. An example global variable file `fsx_vars_example.yml` is included to demonstrate typical configuration. Following are key considerations:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Source/Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster name</td>
<td>source/destination</td>
</tr>
<tr>
<td>cluster management IP</td>
<td>source/destination</td>
</tr>
<tr>
<td>inter-cluster IP</td>
<td>source/destination</td>
</tr>
<tr>
<td>vserver name</td>
<td>source/destination</td>
</tr>
<tr>
<td>vserver management IP</td>
<td>source/destination</td>
</tr>
<tr>
<td>NFS lifs</td>
<td>source/destination</td>
</tr>
<tr>
<td>cluster credentials</td>
<td>fsxadmin and vsadmin pwd to be updated in roles/ontap_setup/defaults/main.yml file</td>
</tr>
</tbody>
</table>

Oracle database volumes – they should have been created from AWS FSx console, volume naming should follow strictly with following standard:

- Oracle binary: `{{ host_name }}_bin`, generally one lun/volume
- Oracle data: `{{ host_name }}_data`, can be multiple luns/volume, add additional line for each additional lun/volume in variable such as `{{ host_name }}_data_01`, `{{ host_name }}_data_02` ...
- Oracle log: `{{ host_name }}_log`, can be multiple luns/volume, add additional line for each additional lun/volume in variable such as `{{ host_name }}_log_01`, `{{ host_name }}_log_02` ...

- host_name: as defined in hosts file in root directory, the code is written to be specifically matched up with host name defined in host file.

Linux and DB specific global variables – keep it as is.

Enter redhat subscription if you have one, otherwise leave it black.

---

Host variables configuration

---

1520
Host variables are defined in host_vars directory named as {{ host_name }}.yml. An example host variable file host_name.yml is included to demonstrate typical configuration. Following are key considerations:

Oracle - define host specific variables when deploying Oracle in multiple hosts concurrently
- ansible_host: IP address of database server host
- log_archive_mode: enable archive log archiving (true) or not (false)
- oracle_sid: Oracle instance identifier
- pdb: Oracle in a container configuration, name pdb_name string and number of pdbs (Oracle allows 3 pdbs free of multitenant license fee)
- listener_port: Oracle listener port, default 1521
- memory_limit: set Oracle SGA size, normally up to 75% RAM
- host_datastores_nfs: combining of all Oracle volumes (binary, data, and log) as defined in global vars file. If multi luns/volumes, keep exactly the same number of luns/volumes in host_var file

Linux - define host specific variables at Linux level
- hugepages_nr: set hugepage for large DB with large SGA for performance
- swap_blocks: add swap space to EC2 instance. If swap exist, it will be ignored.

DB server host file configuration

AWS EC2 instance use IP address for host naming by default. If you use different name in hosts file for Ansible, setup host naming resolution in /etc/hosts file for both source and target servers. Following is an example.

```
127.0.0.1 localhost localhost.localdomain localhost4
localhost4.localdomain4
::1 localhost localhost.localdomain localhost6
localhost6.localdomain6
172.30.15.96 db1
172.30.15.107 db2
```

Playbook execution - executed in sequence
1. Install Ansible controller prerequisites.

```bash
ansible-playbook -i hosts requirements.yml

ansible-galaxy collection install -r collections/requirements.yml --force
```

2. Setup target EC2 DB instance.

```bash
ansible-playbook -i hosts ora_dr_setup.yml -u ec2-user --private-key db2.pem -e @vars/fsx_vars.yml
```

3. Setup FSx ONTAP snapmirror relationship between source and target database volumes.

```bash
ansible-playbook -i hosts ontap_setup.yml -u ec2-user --private-key db2.pem -e @vars/fsx_vars.yml
```

4. Backup Oracle database data volumes via snapshot from crontab.

```bash
10 * * * * cd /home/admin/na_ora_hadr_failover_resync &&
/usr/bin/ansible-playbook -i hosts ora_replication_cg.yml -u ec2-
user --private-key db1.pem -e @vars/fsx_vars.yml >>
logs/snap_data_`date +"%Y-%m%d-%H%M%S"`.log 2>&1
```

5. Backup Oracle database archive log volumes via snapshot from crontab.

```bash
0,20,30,40,50 * * * * cd /home/admin/na_ora_hadr_failover_resync &&
/usr/bin/ansible-playbook -i hosts ora_replication_logs.yml -u ec2-
user --private-key db1.pem -e @vars/fsx_vars.yml >>
logs/snap_log_`date +"%Y-%m%d-%H%M%S"`.log 2>&1
```

6. Run failover and recover Oracle database on target EC2 DB instance - test and validate HA/DR configuration.

```bash
ansible-playbook -i hosts ora_recovery.yml -u ec2-user --private-key db2.pem -e @vars/fsx_vars.yml
```

7. Run resync after failover test - re-establish database volumes snapmirror relationship in replication mode.
ansible-playbook -i hosts ontap_ora_resync.yml -u ec2-user --private -key db2.pem -e @vars/fsx_vars.yml

Where to find additional information

To learn more about the NetApp solution automation, review the following website NetApp Solution Automation

AWS FSx ONTAP Cluster and EC2 Instance Provision

NetApp Solutions Engineering Team

Purpose

This toolkit automates the tasks of provisioning of an AWS FSx ONTAP storage cluster and an EC2 compute instance, which can be subsequently used for database deployment.

This solution addresses the following use cases:

- Provision an EC2 compute instance in AWS cloud in a predefined VPC subnet and set ssh key for EC2 instance access as ec2-user.
- Provision an AWS FSx ONTAP storage cluster in desired availability zones and configure a storage SVM and set cluster admin user fsxadmin password.

Audience

This solution is intended for the following people:

- A DBA who manages databases in AWS EC2 environment.
- A database solution architect who is interested in database deployment in AWS EC2 ecosystem.
- A storage administrator who manages AWS FSx ONTAP storage that supports databases.
- An application owner who likes to standup database in AWS EC2 ecosystem.

License

By accessing, downloading, installing or using the content in this GitHub repository, you agree the terms of the License laid out in License file.

There are certain restrictions around producing and/or sharing any derivative works with the content in this GitHub repository. Please make sure you read the terms of the License before using the content. If you do not agree to all of the terms, do not access, download or use the content in this repository.

Solution deployment

Prerequisites for deployment
Deployment requires the following prerequisites.

- An Organization and AWS account has been setup in AWS public cloud
- An user to run the deployment has been created
- IAM roles has been configured
- IAM roles granted to user to permit provisioning the resources

VPC and security configuration

- A VPC has been created to host the resources to be provisioned
- A security group has been configured for the VPC
- A ssh key pair has been created for EC2 instance access

Network configuration

- Subnets has been created for VPC with network segments assigned
- Route tables and network ACL configured
- NAT gateways or internet gateways configured for internet access

Download the toolkit

```
git clone https://github.com/NetApp/na_aws_fsx_ec2_deploy.git
```

Connectivity and authentication

The toolkit is supposed to be executed from an AWS cloud shell. AWS cloud shell is a browser-based shell that makes it easy to securely manage, explore, and interact with your AWS resources. CloudShell is pre-authenticated with your console credentials. Common development and operations tools are pre-installed, so no local installation or configuration is required.

Terraform provider.tf and main.tf files configuration
The provider.tf defines the provider that Terraform is provisioning resources from via API calls. The main.tf defines the resources and attributes of resources that are to be provisioned. Following are some details:

provider.tf:
```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 4.54.0"
    }
  }
}
```

main.tf:
```
resource "aws_instance" "ora_01" {
  ami                           = var.ami
  instance_type                 = var.instance_type
  subnet_id                     = var.subnet_id
  key_name                      = var.ssh_key_name
  root_block_device {
    volume_type                 = "gp3"
    volume_size                 = var.root_volume_size
  }
  tags = {
    Name                        = var.ec2_tag
  }
}
```

Terraform variables.tf and terraform.tfvars configuration
The variables.tf declares the variables to be used in main.tf. The terraform.tfvars contains the actual values for the variables. Following are some examples:

variables.tf:

```terraform
### EC2 instance variables ###

variable "ami" {
  type        = string
  description = "EC2 AMI image to be deployed"
}

variable "instance_type" {
  type        = string
  description = "EC2 instance type"
}
```

terraform.tfvars:

```terraform
# EC2 instance variables

ami                     = "ami-06640050dc3f556bb" //RedHat 8.6 AMI
instance_type           = "t2.micro"
ec2_tag                 = "ora_01"
subnet_id               = "subnet-04f5fe7073ff514fb"
ssh_key_name            = "sufi_new"
root_volume_size        = 30
```

Step by step procedures - executed in sequence
1. Install Terraform in AWS cloud shell.

```bash
git clone https://github.com/tfutils/tfenv.git ~/.tfenv

mkdir ~/bin

ln -s ~/.tfenv/bin/* ~/bin/

tfenv install

tfenv use 1.3.9
```

2. Download the toolkit from NetApp GitHub public site

```bash
git clone https://github.com/NetApp-Automation/na_aws_fsx_ec2_deploy.git
```

3. Run init to initialize terraform

```bash
terraform init
```

4. Output the execution plan

```bash
terraform plan -out=main.plan
```

5. Apply the execution plan

```bash
terraform apply "main.plan"
```

6. Run destroy to remove the resources when done

```bash
terraform destroy
```
Where to find additional information

To learn more about the NetApp solution automation, review the following website NetApp Solution Automation
Data scientists and engineers often need to access data stored in the NFS format, but accessing this data directly from the S3 protocol in AWS SageMaker can be challenging because AWS only supports S3 bucket access. However, NetApp ONTAP provides a solution by enabling dual-protocol access for NFS and S3. With this solution, data scientists and engineers can access NFS data from AWS SageMaker notebooks via S3 buckets from NetApp Cloud Volumes ONTAP. This approach enables easy access and sharing of the same data from both NFS and S3 without the need for additional software.

Solution technology

This solution utilizes the following technologies:

- **AWS SageMaker Notebook.** Offers machine learning capabilities to developers and data scientists to create, train, and deploy high-quality ML models efficiently.

- **NetApp BlueXP.** Enables the discovery, deployment, and operation of storage on premises as well as on AWS, Azure, and Google Cloud. It provides data protection against data loss, cyber threats, and unplanned outages and optimizes data storage and infrastructure.

- **NetApp Cloud Volumes ONTAP.** Provides enterprise-grade storage volumes with NFS, SMB/CIFS, iSCSI, and S3 protocols on AWS, Azure, and Google Cloud, giving users greater flexibility in accessing and managing their data in the cloud.

NetApp Cloud Volumes ONTAP created from BlueXP to store ML data.

The following figure shows the technical components of the solution.
Use case summary

A potential use case for dual protocol access of NFS and S3 is in the fields of machine learning and data science. For example, a team of data scientists might be working on a machine learning project using AWS SageMaker, which requires access to data stored in the NFS format. However, the data might also need to be accessed and shared via S3 buckets to collaborate with other team members or to integrate with other applications that use S3.

By utilizing NetApp Cloud Volumes ONTAP, the team can store their data in a single location and have it accessible with both NFS and S3 protocols. The data scientists can access the data in NFS format directly from AWS SageMaker, while other team members or applications can access the same data via S3 buckets.

This approach enables the data to be accessed and shared easily and efficiently without the need for additional software or data migration between different storage solutions. It also allows for a more streamlined workflow and collaboration among team members, resulting in faster and more effective development of machine learning models.

Data duality for data scientists and other applications

Data is available in NFS and accessed from S3 from AWS SageMaker.

Technology requirements

You need NetApp BlueXP, NetApp Cloud Volumes ONTAP, and AWS SageMaker Notebooks for the data-duality use case.

Software requirements

The following table lists the software components that are required to implement the use case.

<table>
<thead>
<tr>
<th>Software</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlueXP</td>
<td>1</td>
</tr>
<tr>
<td>NetApp Cloud Volumes ONTAP</td>
<td>1</td>
</tr>
<tr>
<td>AWS SageMaker Notebook</td>
<td>1</td>
</tr>
</tbody>
</table>

Deployment procedures

Deploying the data-duality solution involves the following tasks:

- BlueXP Connector
- NetApp Cloud Volumes ONTAP
- Data for machine learning
- AWS SageMaker
- Validated machine learning from Jupyter Notebooks

BlueXP connector

In this validation, we used AWS. It's also applicable for Azure and Google Cloud. To create a BlueXP Connector in AWS, complete the following steps:
1. We used the credentials based on the mcarl-marketplace-subscription in BlueXP.

2. Choose the region suitable for your environment (for example, us-east-1 [N. Virginia]), and select the authentication method (for example, Assume Role or AWS keys). In this validation, we use AWS keys.

3. Provide the name of the connector and create a role.

4. Provide the network details such as the VPC, subnet, or keypair, depending on whether you need a public IP or not.

5. Provide the details for the security group, such as HTTP, HTTPS, or SSH access from the source type, such as anywhere and IP range information.

6. Review and create the BlueXP Connector.

7. Verify that the BlueXP EC2 instance state is running in the AWS console, and check the IP address from the Networking tab.

8. Log into the connector user interface from the BlueXP portal, or you can use the IP address for access from the browser.

NetApp Cloud Volumes ONTAP

To create a Cloud Volumes ONTAP instance in BlueXP, complete the following steps:

1. Create a new working environment, select the cloud provider, and select the type of Cloud Volumes ONTAP instance, (such as single-CVO, HA, or Amazon FSxN for ONTAP).

2. Provide details such as the Cloud Volumes ONTAP cluster name and credentials. In this validation, we created a Cloud Volumes ONTAP instance called svm_sagemaker_cvo_sn1.

3. Select the services needed for Cloud Volumes ONTAP. In this validation, we choose to only monitor, so we disabled Data Sense & Compliance and Backup to Cloud Services.

4. In the Location & Connectivity section, select the AWS region, VPC, subnet, security group, SSH authentication method, and either a password or a key pair.

5. Choose the charging method. We used Professional for this validation.

6. You can choose a preconfigured package, such as POC and Small Workloads, Database and Application Data Production Workloads, Cost Effective DR, or Highest Performance Production Workloads. In this validation, we choose Poc and Small Workloads.

7. Create a volume with a specific size, allowed protocols, and export options. In this validation, we created a volume called vol1.

8. Choose a profile disk type and tiering policy. In this validation, we disabled Storage Efficiency and General- Purpose SSD – Dynamic Performance.

9. Finally, review and create the Cloud Volumes ONTAP instance. Then wait for 15-20 minutes for BlueXP to create the Cloud Volumes ONTAP working environment.

10. Configure the following parameters to enable the Duality protocol. The Duality protocol (NFS/S3) is supported from ONTAP 9.12.1 and later.
    a. In this validation, we created an SVM called svm_sagemaker_cvo_sn1 and volume vol1.
    b. Verify that the SVM has the protocol support for NFS and S3. If not, modify the SVM to support them.
sagemaker_cvo_sn1::> vserver show -vserver svm_sagemaker_cvo_sn1
Vserver: svm_sagemaker_cvo_sn1
  Vserver Type: data
  Vserver Subtype: default
  Vserver UUID: 911065dd-a8bc-11ed-bc24-e1c0f00ad86b
  Root Volume:

svm_sagemaker_cvo_sn1_root
  Aggregate: aggr1
  NIS Domain: -
  Root Volume Security Style: unix
  LDAP Client: -
  Default Volume Language Code: C.UTF-8
  Snapshot Policy: default
  Data Services: data-cifs, data-flexcache,
                 data-iscsi, data-nfs,
                 data-nvme-tcp
  Comment:
  Quota Policy: default
  List of Aggregates Assigned: aggr1
  Limit on Maximum Number of Volumes allowed: unlimited
  Vserver Admin State: running
  Vserver Operational State: running
  Vserver Operational State Stopped Reason: -
    Allowed Protocols: nfs, cifs, fcp, iscsi, ndmp, s3
  Disallowed Protocols: nvme
  Is Vserver with Infinite Volume: false
  QoS Policy Group: -
  Caching Policy Name: -
    Config Lock: false
    IPspace Name: Default
  Foreground Process: -
    Logical Space Reporting: true
    Logical Space Enforcement: false
  Default Anti_ransomware State of the Vserver's Volumes: disabled
  Enable Analytics on New Volumes: false
  Enable Activity Tracking on New Volumes: false

sagemaker_cvo_sn1::>

11. Create and install a CA certificate if required.
12. Create a service data policy.
sagemaker_cvo_sn1::*> network interface service-policy create -vserver svm_sagemaker_cvo_sn1 -policy sagemaker_s3_nfs_policy -services data-core, data-s3-server, data-nfs, data-flexcache
sagemaker_cvo_sn1::*> network interface create -vserver svm_sagemaker_cvo_sn1 -lif svm_sagemaker_cvo_sn1_s3_lif -service-policy sagemaker_s3_nfs_policy -home-node sagemaker_cvo_sn1-01 -address 172.30.10.41 -netmask 255.255.255.192

Warning: The configured failover-group has no valid failover targets for the LIF's failover-policy. To view the failover targets for a LIF, use the "network interface show -failover" command.

sagemaker_cvo_sn1::*> network interface show
Logical   Status     Network            Current       Current Is
Vserver   Interface  Admin/Oper Address/Mask       Node          Port
Home
----------- ---------- ---------- ------------------ --------------
------- ---- ------- ------
sagemaker_cvo_sn1
  cluster-mgmt up/up  172.30.10.40/26  sagemaker_cvo_sn1-
  01
  true
  intercluster up/up  172.30.10.48/26  sagemaker_cvo_sn1-
  01
  true
  svm_sagemaker_cvo_sn1
    svm_sagemaker_cvo_sn1_data_lif
       up/up  172.30.10.23/26  sagemaker_cvo_sn1-
       01
       true
    svm_sagemaker_cvo_sn1_mgmt_lif
       up/up  172.30.10.32/26  sagemaker_cvo_sn1-
       01
       true
    svm_sagemaker_cvo_sn1_s3_lif
       up/up  172.30.10.41/26  sagemaker_cvo_sn1-
true
6 entries were displayed.

sagemaker_cvo_sn1::*>
sagemaker_cvo_sn1::*> vserver object-store-server create -vserver
svm_sagemaker_cvo_sn1  -is-http-enabled true -object-store-server
svm_sagemaker_cvo_s3_sn1 -is-https-enabled false
sagemaker_cvo_sn1::*> vserver object-store-server show

Vserver: svm_sagemaker_cvo_sn1

Object Store Server Name: svm_sagemaker_cvo_s3_sn1
  Administrative State: up
    HTTP Enabled: true
  Listener Port For HTTP: 80
    HTTPS Enabled: false
  Secure Listener Port For HTTPS: 443
  Certificate for HTTPS Connections: -
    Default UNIX User: pcuser
    Default Windows User: -
  Comment:

sagemaker_cvo_sn1::*>

13. Check the aggregate details.
sagemaker_cvo_sn1:*> aggr show

<table>
<thead>
<tr>
<th>Aggregate</th>
<th>Size</th>
<th>Available</th>
<th>Used%</th>
<th>State</th>
<th>#Vols</th>
<th>Nodes</th>
<th>RAID</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggr0_sagemaker_cvo_sn1_01</td>
<td>124.0GB</td>
<td>50.88GB</td>
<td>59%</td>
<td>online</td>
<td>1</td>
<td>sagemaker_cvo_raid0, sn1-01</td>
<td>normal</td>
</tr>
<tr>
<td>aggr1</td>
<td>907.1GB</td>
<td>904.9GB</td>
<td>0%</td>
<td>online</td>
<td>2</td>
<td>sagemaker_cvo_raid0, sn1-01</td>
<td>normal</td>
</tr>
</tbody>
</table>

2 entries were displayed.

sagemaker_cvo_sn1:*> 14. Create a user and group.
15. Create a bucket on the NFS volume.
**AWS SageMaker**

To create an AWS Notebook from AWS SageMaker, complete the following steps:

1. Make sure the user who is creating Notebook instance has an AmazonSageMakerFullAccess IAM policy or is part of an existing group that has AmazonSageMakerFullAccess rights. In this validation, the user is part of an existing group.

2. Provide the following information:
   - Notebook instance name.
   - Instance type.
   - Platform identifier.
   - Select the IAM role that has AmazonSageMakerFullAccess rights.
   - Root access – enable.
   - Encryption key - Select no custom encryption.
   - Keep the remaining default options.

3. In this validation, the SageMaker instance details are as follows:
4. Start the AWS Notebook.

5. Open the Jupyter lab.
6. Log into the terminal and mount the Cloud Volumes ONTAP volume.

```
sh-4.2$ sudo mkdir /vol1; sudo mount -t nfs 172.30.10.41:/vol1 /vol1
sh-4.2$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        2.0G   0   2.0G   0% /dev
tmpfs            2.0G   0   2.0G   0% /dev/shm
tmpfs            2.0G   624K  2.0G   1% /run
tmpfs            2.0G   0   2.0G   0% /sys/fs/cgroup
/dev/xvda1       140G  114G   27G  82% /
/dev/xvdf        4.8G   72K  4.6G   1% /home/ec2-user/SageMaker
tmpfs            393M   0   393M   0% /run/user/1001
tmpfs            393M   0   393M   0% /run/user/1002
tmpfs            393M   0   393M   0% /run/user/1000
172.30.10.41:/vol1 973M  189M  785M  20% /vol1
sh-4.2$
```

7. Check the bucket created on the Cloud Volumes ONTAP volume using the AWS CLI commands.
Data for machine learning

In this validation, we used a dataset from DBpedia, a crowd-sourced community effort, to extract structured content from the information created in various Wikimedia projects.

1. Download the data from the DBpedia GitHub location and extract it. Use the same terminal used in the previous section.
2. Copy the data to the Cloud Volumes ONTAP location and check it from the S3 bucket using the AWS CLI.
sh-4.2$ df -h
Filesystem          Size  Used Avail Use% Mounted on
devtmpfs            2.0G     0  2.0G   0% /dev
tmpfs               2.0G     0  2.0G   0% /dev/shm
tmpfs               2.0G  628K  2.0G   1% /run
tmpfs               2.0G     0  2.0G   0% /sys/fs/cgroup
/dev/xvda1          140G  114G   27G  82% /
/dev/xvdf           4.8G  52K  4.6G   1% /home/ec2-user/SageMaker
tmpfs               393M     0  393M   0% /run/user/1002
tmpfs               393M     0  393M   0% /run/user/1001
tmpfs               393M     0  393M   0% /run/user/1000
172.30.10.41:/voll  973M  384K  973M   1% /voll
sh-4.2$ pwd
/home/ec2-user
sh-4.2$ cp -ra dbpedia_csv /voll
sh-4.2$ aws s3 ls --profile netapp --endpoint-url s3://ontapbucket1/
  PRE dbpedia_csv/
2023-02-10 18:46:44       4747 1
2023-02-10 18:48:32         96 setup.cfg
sh-4.2$

3. Perform basic validation to make sure that read/write functionality works on the S3 bucket.

sh-4.2$ aws s3 cp --profile netapp --endpoint-url /usr/share/doc/util-linux-2.30.2 s3://ontapbucket1/ --recursive
upload: ../../../usr/share/doc/util-linux-2.30.2/deprecated.txt to s3://ontapbucket1/deprecated.txt
upload: ../../../usr/share/doc/util-linux-2.30.2/getopt-parse.bash to s3://ontapbucket1/getopt-parse.bash
upload: ../../../usr/share/doc/util-linux-2.30.2/README to s3://ontapbucket1/README
upload: ../../../usr/share/doc/util-linux-2.30.2/getopt-parse.tcsh to s3://ontapbucket1/getopt-parse.tcsh
upload: ../../../usr/share/doc/util-linux-2.30.2/AUTHORS to s3://ontapbucket1/AUTHORS
upload: ../../../usr/share/doc/util-linux-2.30.2/NEWS to s3://ontapbucket1/NEWS
sh-4.2$ aws s3 ls --profile netapp --endpoint-url s3://ontapbucket1/s3://ontapbucket1/
An error occurred (InternalError) when calling the ListObjectsV2 operation: We encountered an internal error. Please try again.
sh-4.2$ aws s3 ls --profile netapp --endpoint-url s3://ontapbucket1/
  PRE dbpedia_csv/
  2023-02-16 19:19:27      26774 AUTHORS
sh-4.2$ ls -ltr /vol1
total 132
  drwxrwxr-x  2 ec2-user ec2-user  4096 Mar 29  2015 dbpedia_csv
 -rw-r--r--  1 nobody nobody    2245 Apr 10 17:37 getopt-parse.tcsh
 -rw-r--r--  1 nobody nobody    2825 Apr 10 17:37 deprecated.txt
 -rw-r--r--  1 nobody nobody    4493 Apr 10 17:37 README
 -rw-r--r--  1 nobody nobody    1590 Apr 10 17:37 getopt-parse.bash
 -rw-r--r--  1 nobody nobody    26774 Apr 10 17:37 AUTHORS
 -rw-r--r--  1 nobody nobody    72727 Apr 10 17:37 NEWS
sh-4.2$ ls -ltr /vol1/dbpedia_csv/
total 192104
 -rw-------  1 ec2-user ec2-user 174148970 Mar 28  2015 train.csv
 -rw-------  1 ec2-user ec2-user  21775285 Mar 28  2015 test.csv
 -rw-------  1 ec2-user ec2-user       146 Mar 28  2015 classes.txt
 -rw-rw-r--  1 ec2-user ec2-user      1758 Mar 29  2015 readme.txt
sh-4.2$ chmod -R 777 /vol1/dbpedia_csv
sh-4.2$ ls -ltr /vol1/dbpedia_csv/
total 192104
 -rwxrwxrwx  1 ec2-user ec2-user 174148970 Mar 28  2015 train.csv
 -rwxrwxrwx  1 ec2-user ec2-user  21775285 Mar 28  2015 test.csv
 -rwxrwxrwx  1 ec2-user ec2-user       146 Mar 28  2015 classes.txt
 -rwxrwxrwx  1 ec2-user ec2-user      1758 Mar 29  2015 readme.txt
sh-4.2$ aws s3 cp --profile netapp --endpoint-url http://172.30.2.248/s3://ontapbucket1/ /tmp --recursive
download: s3://ontapbucket1/AUTHORS to ../../tmp/AUTHORS
download: s3://ontapbucket1/README to ../../tmp/README
download: s3://ontapbucket1/NEWS to ../../tmp/NEWS
download: s3://ontapbucket1/dbpedia_csv/classes.txt to
   ../../tmp/dbpedia_csv/classes.txt
download: s3://ontapbucket1/dbpedia_csv/readme.txt to
   ../../tmp/dbpedia_csv/readme.txt
download: s3://ontapbucket1/deprecated.txt to ../../tmp/deprecated.txt
download: s3://ontapbucket1/getopt-parse.bash to ../../tmp/getopt-parse.bash
download: s3://ontapbucket1/getopt-parse.tcsh to ../../tmp/getopt-parse.tcsh
download: s3://ontapbucket1/dbpedia_csv/test.csv to
   ../../tmp/dbpedia_csv/test.csv
download: s3://ontapbucket1/dbpedia_csv/train.csv to
   ../../tmp/dbpedia_csv/train.csv
sh-4.2$
Validate machine learning from Jupyter Notebooks

The following validation provides the machine-learning build, train, and deploy models through text classification by using the SageMaker BlazingText example below:

1. Install the boto3 and SageMaker packages.

In [1]: pip install --upgrade boto3 sagemaker

Output:

Looking in indexes: https://pypi.org/simple,
https://pip.repos.neuron.amazonaws.com
Collecting boto3
  Downloading boto3-1.26.72-py3-none-any.whl (132 kB)
    132.7/132.7 kB 14.6 MB/s eta 0:00:00
Requirement already satisfied: sagemaker in /home/ec2-
  Requirement already satisfied: botocore<1.30.0,>=1.29.72 in /home/ec2-
  Requirement already satisfied: s3transfer<0.7.0,>=0.6.0 in /home/ec2-

1544
Building wheels for collected packages: sagemaker

Building wheel for sagemaker (setup.py) ... done

Created wheel for sagemaker: filename=sagemaker-2.132.0-py2.py3-none-any. whl size=905449
sha256=f6100a5dc95627f2e2a49824e38f0481459a27805ee19b5a06ec
83db0252fd41

Stored in directory: /home/ec2-user/.cache/pip/wheels/60/41/b6/482e7ab096
520df034f8f2d7d244a1d7ba0681b27ef45aa61

Successfully built sagemaker

Installing collected packages: botocore, boto3, sagemaker

Attempting uninstall: botocore        Found existing installation: botocore 1.24.19

    Uninstalling botocore-1.24.19:   Successfully uninstalled botocore-1.24.19

Attempting uninstall: boto3           Found existing installation: boto3 1.26.44

    Uninstalling boto3-1.26.44:      Successfully uninstalled boto3-1.26.44

Attempting uninstall: sagemaker       Found existing installation: sagemaker 2.127.0
Uninstalling sagemaker-2.127.0:
  Successfully uninstalled sagemaker-2.127.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
awscli 1.27.44 requires botocore==1.29.44, but you have botocore 1.29.72 which is incompatible.
aiobotocore 2.0.1 requires botocore<1.22.9,>=1.22.8, but you have botocore 1.29.72 which is incompatible. Successfully installed boto3-1.26.72 botocore-1.29.72 sagemaker-2.132.0 Note: you may need to restart the kernel to use updated packages.

2. In the following step, the data (dbpedia_csv) is downloaded from the s3 bucket ontapbucket1 to a Jupyter Notebook instance used in machine learning.
In [2]: import sagemaker
In [3]: from sagemaker import get_execution_role
In [4]: import json
import boto3
sess = sagemaker.Session()
role = get_execution_role()
print(role)
bucket = "ontapbucket1"
print(bucket)
sess.s3_client = boto3.client('s3', region_name='', aws_access_key_id = '0ZNAX21JW5Q8AP80CQ2E', aws_secret_access_key = 'PpLs4gA9K0_2gPhuykkp014gBjcC9Rbi3QDX_6rr', use_ssl = False, endpoint_url = 'http://172.30.10.41',
    config=boto3.session.Config(signature_version='s3v4', s3={'addressing_style':'path'}))
sess.s3_resource = boto3.resource('s3', region_name='', aws_access_key_id = '0ZNAX21JW5Q8AP80CQ2E', aws_secret_access_key = 'PpLs4gA9K0_2gPhuykkp014gBjcC9Rbi3QDX_6rr', use_ssl = False, endpoint_url = 'http://172.30.10.41',
    config=boto3.session.Config(signature_version='s3v4', s3={'addressing_style':'path'}))
prefix = "blazingtext/supervised"
import os
my_bucket = sess.s3_resource.Bucket(bucket)
#os.mkdir('dbpedia_csv')
for s3_object in my_bucket.objects.all():
    filename = s3_object.key
    # print(filename)
    # print(s3_object.key)
    my_bucket.download_file(s3_object.key, filename)

3. The following code creates the mapping from integer indices to class labels that are used to retrieve the actual class name during inference.

```python
index_to_label = {}
with open("dbpedia_csv/classes.txt") as f:
    for i, label in enumerate(f.readlines()):
        index_to_label[str(i + 1)] = label.strip()
```
The output lists the files and folders in the ontapbucket1 bucket that are used as data for the AWS SageMaker machine-learning validation.

4. Start the data preprocessing phase to preprocess the training data into a space-separated, tokenized text format that can be consumed by the BlazingText algorithm and the nltk library to tokenize the input sentences from the DBPedia dataset. Download the nltk tokenizer and other libraries. The `transform_instance` applied to each data instance in parallel uses the Python multiprocessing module.

```python
In [7]: from random import shuffle
import multiprocessing
from multiprocessing import Pool
import csv
import nltk
nltk.download("punkt")
def transform_instance(row):
    cur_row = []
    label = "__label__" + index_to_label [row[0]] # Prefix the index-ed label with __label__
    cur_row.append (label)
    cur_row.extend(nltk.word_tokenize(row[1].lower ()))
    cur_row.extend(nltk.word_tokenize(row[2].lower ()))
    return cur_row

def preprocess(input_file, output_file, keep=1):
```
```python
all_rows = []
with open(input_file, "r") as csvinfile:
    csv_reader = csv.reader(csvinfile, delimiter="","")
    for row in csv_reader:
        all_rows.append(row)
shuffle(all_rows)
all_rows = all_rows[: int(keep * len(all_rows))]
pool = Pool(processes=multiprocessing.cpu_count())
transformed_rows = pool.map(transform_instance, all_rows)
pool.close()
pool.join()
with open(output_file, "w") as csvoutfile:
    csv_writer = csv.writer(csvoutfile, delimiter=" ",
                            lineterminator="\n")
    csv_writer.writerows(transformed_rows)

# Preparing the training dataset
# since preprocessing the whole dataset might take a couple of minutes,
# we keep 20% of the training dataset for this demo.
# Set keep to 1 if you want to use the complete dataset
preprocess("dbpedia_csv/train.csv","dbpedia.train", keep=0.2)
# Preparing the validation dataset
preprocess("dbpedia_csv/test.csv","dbpedia.validation")
sess = sagemaker.Session()
role = get_execution_role()
print (role) # This is the role that sageMaker would use to leverage AWS
resources (S3, Cloudwatch) on your behalf
bucket = sess.default_bucket() # Replace with your own bucket name if
needed
print("default Bucket::: ")
print(bucket)
```

Output:

```
[nltk_data] Downloading package punkt to /home/ec2-user/nltk_data...  
[nltk_data] Package punkt is already up-to-date!
arn:aws:iam::210811600188:role/SageMakerFullRole default Bucket:::  
sagemaker-us-east-1-210811600188
```

5. Upload the formatted and training dataset to S3 so that it can be used by SageMaker to execute training jobs. Then upload two files to the bucket and prefix location using the Python SDK.
ln [8]: %time
train_channel = prefix + "/train"
validation_channel = prefix + "/validation"
sess.upload_data(path="dbpedia.train", bucket=bucket,
key_prefix=train_channel)
sess.upload_data(path="dbpedia.validation", bucket=bucket,
key_prefix=validation_channel)
s3_train_data = "s3://{}".format(bucket, train_channel)
s3_validation_data = "s3://{}".format(bucket, validation_channel)

Output:

CPU times: user 546 ms, sys: 163 ms, total: 709 ms
Wall time: 1.32 s

6. Set up an output location at S3 where the model artifact is loaded so that artifacts can be the output of the
algorithm’s training job. Create a sageMaker.estimator.Estimator object to launch the training job.

In [9]: s3_output_location = "s3://{}".format(bucket, prefix)
In [10]: region_name = boto3.Session().region_name
In [11]: container =
sagemaker.amazon.amazon_estimator.get_image_uri(region_name,
"blazingtext","latest")
print("Using SageMaker BlazingText container: {} ({}).format(container,
region_name))

Output:

The method get_image_uri has been renamed in sagemaker>=2.
Defaulting to the only supported framework/algorithm version: 1.
Ignoring framework/algorithm version: latest.
Using SageMaker BlazingText container: 811284229777.dkr.ecr.us-east-1.amazonaws.com/blazingtext:1 (us-east-1)

7. Define the SageMaker Estimator with resource configurations and hyperparameters to train text
classification on the DBPedia dataset using the supervised mode on a c4.4xlarge instance.
In [12]: bt_model = sagemaker.estimator.Estimator(
    container,
    role,
    instance_count=1,
    instance_type="ml.c4.4xlarge",
    volume_size=30,
    max_run=360000,
    input_mode="File",
    output_path=s3_output_location,
    hyperparameters={
        "mode": "supervised",
        "epochs": 1,
        "min_count": 2,
        "learning_rate": 0.05,
        "vector_dim": 10,
        "early_stopping": True,
        "patience": 4,
        "min_epochs": 5,
        "word_ngrams": 2,
    },
)

8. Prepare a handshake between the data channels and the algorithm. To do this, create the
sagemaker.session.s3_input objects from the data channels, and keep them in a dictionary for the
algorithm to consume.

In [13]: train_data = sagemaker.inputs.TrainingInput(
    s3_train_data,
    distribution="FullyReplicated",
    content_type="text/plain",
    s3_data_type="S3Prefix",
)
validation_data = sagemaker.inputs.TrainingInput(
    s3_validation_data,
    distribution="FullyReplicated",
    content_type="text/plain",
    s3_data_type="S3Prefix",
)
data_channels = {"train": train_data, "validation": validation_data}

9. After the job has finished, a Job Complete message appears. The trained model can be found in the S3
bucket that was set up as the output_path in the estimator.
ln [14]: bt_model.fit(inputs=data_channels, logs=True)

Output:

INFO:sagemaker:Creating training-job with name: blazingtext-2023-02-16-20-3
7-30-748
2023-02-16 20:37:30 Starting - Starting the training job......
2023-02-16 20:38:09 Starting - Preparing the instances for training......
2023-02-16 20:39:24 Downloading - Downloading input data
2023-02-16 20:39:24 Training - Training image download completed.
Training in progress... Arguments: train
[02/16/2023 20:39:41 WARNING 140279908747072] Loggers have already been set up. [02/16/2023 20:39:41 WARNING 140279908747072] Loggers have already been set up.
[02/16/2023 20:39:41 INFO 140279908747072] nvidia-smi took: 0.0251793861389
16016 secs to identify 0 gpus
Number of CPU sockets found in instance is 1
Read 6M words
Number of words: 149301
Loading validation data from /opt/ml/input/data/validation/dbpedia.validation on
Loaded validation data.
-------------- End of epoch: 1 ##### Alpha: 0.0000  Progress: 100.00%
Million Words/sec: 10.39 ##### Training finished.
Average throughput in Million words/sec: 10.39
Total training time in seconds: 0.60
#train_accuracy: 0.7223
Number of train examples: 112000
#validation_accuracy: 0.7205
Number of validation examples: 70000
2023-02-16 20:39:55 Uploading - Uploading generated training model
2023-02-16 20:40:11 Completed - Training job completed
Training seconds: 68
Billable seconds: 68
10. After training is complete, deploy the trained model as an Amazon SageMaker real-time hosted endpoint to make predictions.

```
In [15]: from sagemaker.serializers import JSONSerializer
text_classifier = bt_model.deploy(
    initial_instance_count=1, instance_type="ml.m4.xlarge",
    serializer=JSONS
)
```

Output:

```
INFO:sagemaker:Creating model with name: blazingtext-2023-02-16-20-41-33-10
0
INFO:sagemaker:Creating endpoint-config with name blazingtext-2023-02-16-20-41-33-100
INFO:sagemaker:Creating endpoint with name blazingtext-2023-02-16-20-41-33-100
--------!
```

```
In [16]: sentences = [
    "Convair was an american aircraft manufacturing company which later expanded into rockets and spacecraft.",
    "Berwick secondary college is situated in the outer melbourne metropolitan suburb of berwick.",
]
# using the same nltk tokenizer that we used during data preparation for training
tokenized_sentences = [" ".join(nltk.word_tokenize(sent)) for sent in sentences]
payload = {"instances": tokenized_sentences} response =
text_classifier.predict(payload)
predictions = json.loads(response)
print(json.dumps(predictions, indent=2))
```
11. By default, the model returns one prediction with the highest probability. To retrieve the top $k$ predictions, set $k$ in the configuration file.

```python
In [17]: payload = {"instances": tokenized_sentences, "configuration": {"k": 2}}
    response = text_classifier.predict(payload)

predictions = json.loads(response)
print(json.dumps(predictions, indent=2))
```
12. Delete the endpoint before closing the notebook.

```
In [18]: sess.delete_endpoint(text_classifier.endpoint)
WARNING:sagemaker.deprecations:The endpoint attribute has been renamed
in sagemaker>=2.
INFO:sagemaker:Deleting endpoint with name: blazingtext-2023-02-16-20-
41-33
-100
```

**Conclusion**

Based on this validation, Data scientists and engineers can access NFS data from AWS SageMaker Jupyter Notebooks via S3 buckets from NetApp Cloud Volumes ONTAP. This approach enables easy access and sharing of the same data from both NFS and S3 without the need for additional software.

**Where to find additional information**

To learn more about the information that is described in this document, review the following documents and/or websites:
Apache Kafka workloads with NetApp NFS storage

TR-4947: Apache Kafka workload with NetApp NFS storage - Functional validation and performance

Shantanu Chakole, Karthikeyan Nagalingam, and Joe Scott, NetApp

Kafka is a distributed publish-subscribe messaging system with a robust queue that can accept large amounts of message data. With Kafka, applications can write and read data to topics in a very fast manner. Because of its fault tolerance and scalability, Kafka is often used in the big data space as a reliable way to ingest and move many data streams very quickly. Use cases include stream processing, website-activity tracking, metrics collection and monitoring, log aggregation, real time analytics, and so on.

Although normal Kafka operations on NFS work well, the silly rename issue crashes the application during the resizing or repartitioning of a Kafka cluster running on NFS. This is a significant issue because a Kafka cluster must be resized or repartitioned for load-balancing or maintenance purposes. You can find additional details here.

This document describes the following subjects:

- The silly-rename problem and solution validation
- Reducing CPU utilization to reduce the I/O wait time
- Faster Kafka broker recovery time
- Performance in the cloud and on-premises

Why use NFS storage for Kafka workloads?

Kafka workloads in production applications can stream huge amounts of data between applications. This data is held and stored in the Kafka broker nodes in the Kafka cluster. Kafka is also known for availability and parallelism, which it achieves by breaking topics into partitions and then replicating those partitions throughout the cluster. This eventually means that the huge amount of data that flows through a Kafka cluster is generally multiplied in size. NFS makes rebalancing data as the number of brokers changes very quick and easy. For large environments, rebalancing data across DAS when the number of brokers changes is very time consuming, and, in most Kafka environments, the number of brokers changes frequently.

Other benefits include the following:

- Maturity. NFS is a mature protocol, which means most aspects of implementing, securing, and using it are well understood.
• **Open.** NFS is an open protocol, and its continued development is documented in internet specifications as a free and open network protocol.

• **Cost-effective.** NFS is a low-cost solution for network file sharing that is easy to set up because it uses the existing network infrastructure.

• **Centrally managed.** Centralized management of NFS decreases the need for added software and disk space on individual user systems.

• **Distributed.** NFS can be used as a distributed file system, reducing the need for removable media storage devices.

### Why NetApp for Kafka workloads?

The NetApp NFS implementation is considered a gold standard for the protocol and is used in countless enterprise NAS environments. In addition to the credibility of NetApp, it also offers the following benefits:

- Reliability and efficiency
- Scalability and performance
- High availability (HA partner in a NetApp ONTAP cluster)
- Data protection
  - **Disaster recovery (NetApp SnapMirror).** Your site goes down or you want to jump start at a different site and continue from where you left off.
  - Manageability of your storage system (administration and management using NetApp OnCommand).
  - **Load balancing.** The cluster allows you to access different volumes from data LIFs hosted on different nodes.
  - **Nondisruptive operations.** LIFs or volume moves are transparent to the NFS clients.

### NetApp solution for silly rename issue for NFS to Kafka workloads

Kafka is built with the assumption that the underlying filesystem is POSIX compliant: for example, XFS or Ext4. Kafka resource rebalancing removes files while the application is still using them. A POSIX-compliant file system allows unlink to proceed. However, it only removes the file after all references to the file are gone. If the underlying filesystem is network attached, then the NFS client intercepts the unlink calls and manages the workflow. Because there are pending opens on the file being unlinked, the NFS client sends a rename request to the NFS server and, on the last close of the unlinked file, issues a remove operation on the renamed file. This behavior is commonly referred to as NFS silly rename, and it is orchestrated by the NFS client.

Any Kafka broker using storage from an NFSv3 server runs into issues because of this behavior. However, the NFSv4.x protocol has features to address this issue by allowing the server to take responsibility for the opened, unlinked files. NFS servers supporting this optional feature communicate the ownership capability to the NFS client at the time of file opening. The NFS client then ceases the unlink management when there are opens pending and allows the server to manage the flow. Although the NFSv4 specification provides guidelines for implementation, until now, there were not any known NFS server implementations that supported this optional feature.

The following changes are required for the NFS server and the NFS client to address the silly rename issue:
• **Changes to NFS client (Linux).** At the time of file opening, the NFS server responds with a flag, indicating the capability to handle the unlinking of opened files. NFS client-side changes allow the NFS server to handle the unlinking in the presence of the flag. NetApp has updated the open-source Linux NFS client with these changes. The updated NFS client is now generally available in RHEL8.7 and RHEL9.1.

• **Changes to NFS server.** The NFS server keeps track of opens. Unlinking on an existing open file is now managed by the server to match POSIX semantics. When the last open is closed, The NFS server then initiates the actual removal of the file and thus avoids the silly rename process. The ONTAP NFS server has implemented this capability in its latest release, ONTAP 9.12.1.

With the above changes to the NFS client and server, Kafka can safely reap all the benefits of network-attached NFS storage.

**Functional validation - Silly rename fix**

For the functional validation, we showed that a Kafka cluster with an NFSv3 mount for storage fails to perform Kafka operations like partition redistribution, whereas another cluster mounted on NFSv4 with the fix can perform the same operations without any disruptions.

**Validation setup**

The setup is run on AWS. The following table shows the different platform components and environmental configuration used for the validation.

<table>
<thead>
<tr>
<th>Platform component</th>
<th>Environment configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confluent Platform version 7.2.1</td>
<td>• 3 x zookeepers – t3.xlarge</td>
</tr>
<tr>
<td></td>
<td>• 4 x broker servers – r3.xlarge</td>
</tr>
<tr>
<td></td>
<td>• 1 x Grafana – t3.xlarge</td>
</tr>
<tr>
<td></td>
<td>• 1 x control center – t3.xlarge</td>
</tr>
<tr>
<td></td>
<td>• 3 x Producer/consumer</td>
</tr>
<tr>
<td>Operating system on all nodes</td>
<td>RHEL8.7 or later</td>
</tr>
<tr>
<td>NetApp Cloud Volumes ONTAP instance</td>
<td>Single-node instance – M5.2xLarge</td>
</tr>
</tbody>
</table>

The following figure show the architectural configuration for this solution.
**Architectural flow**

- **Compute.** We used a four-node Kafka cluster with a three-node zookeeper ensemble running on dedicated servers.
- **Monitoring.** We used two nodes for a Prometheus-Grafana combination.
- **Workload.** For generating workloads, we used a separate three-node cluster that can produce to and consume from this Kafka cluster.
- **Storage.** We used a single-node NetApp Cloud volumes ONTAP instance with two 500GB GP2 AWS-EBS volumes attached to the instance. These volumes were then exposed to the Kafka cluster as single NFSv4.1 volume through a LIF.

The default properties of Kafka were chosen for all servers. The same was done for the zookeeper swarm.

**Methodology of testing**

1. Update `-is-preserve-unlink-enabled true` to the kafka volume, as follows:

   ```bash
   aws-shantanclastrecall-aws::*> volume create -vserver kafka_svm -volume kafka_fg_vol01 -aggregate kafka_aggr -size 3500GB -state online -policy kafka_policy -security-style unix -unix-permissions 0777 -junction-path /kafka_fg_vol01 -type RW -is-preserve-unlink-enabled true
   [Job 32] Job succeeded: Successful
   ```

2. Two similar Kafka clusters were created with the following difference:
- **Cluster 1.** The backend NFS v4.1 server running production-ready ONTAP version 9.12.1 was hosted by a NetApp CVO instance. RHEL 8.7/RHEL 9.1 were installed on the brokers.
- **Cluster 2.** The backend NFS server was a manually created generic Linux NFSv3 server.

3. A demo topic was created on both the Kafka clusters.

**Cluster 1:**

```
[root@ip-172-30-0-160 demo]# kafka-topics --bootstrap-server=172.30.0.160:9092,172.30.0.172:9092,172.30.0.188:9092,172.30.0.198:9092 --describe --topic __a_demo_topic

Topic: __a_demo_topic TopicId: 2ty29xhQl6SHkxQv-pg PartitionCount: 4 ReplicationFactor: 2 Configs:
  min.insync.replacs=1,segment.bytes=1073741824
  Topic: __a_demo_topic Partition: 0 Leader: 4 Replicas: 4,1 Isr: 4,1 Offline:
  Topic: __a_demo_topic Partition: 1 Leader: 3 Replicas: 3,2 Isr: 3,2 Offline:
  Topic: __a_demo_topic Partition: 2 Leader: 2 Replicas: 2,4 Isr: 2,4 Offline:
  Topic: __a_demo_topic Partition: 3 Leader: 1 Replicas: 1,3 Isr: 1,3 Offline:
```

**Cluster 2:**

```
[root@ip-172-30-0-198 demo]# kafka-topics --bootstrap-server=172.30.0.160:9092,172.30.0.172:9092,172.30.0.188:9092,172.30.0.198:9092 --describe --topic __a_demo_topic

Topic: __a_demo_topic TopicId: A0qpsZTQShyMihauuCG3Q PartitionCount: 4 ReplicationFactor: 2 Configs:
  min.insync.replacs=1,segment.bytes=1073741824
  Topic: __a_demo_topic Partition: 0 Leader: 2 Replicas: 2,3 Isr: 2,3 Offline:
  Topic: __a_demo_topic Partition: 1 Leader: 3 Replicas: 3,1 Isr: 3,1 Offline:
  Topic: __a_demo_topic Partition: 2 Leader: 1 Replicas: 1,4 Isr: 1,4 Offline:
  Topic: __a_demo_topic Partition: 3 Leader: 4 Replicas: 4,2 Isr: 4,2 Offline:
```

4. Data was loaded into these newly created topics for both clusters. This was done using the producer-perf-test toolkit that comes in the default Kafka package:

```
```

5. A health check was performed for broker-1 for each of the clusters using telnet:

- `telnet 172.30.0.160 9092`
- `telnet 172.30.0.198 9092`

A successful health check for brokers on both clusters is shown in the next screenshot:
6. To trigger the failure condition that causes Kafka clusters using NFSv3 storage volumes to crash, we initiated the partition reassignment process on both clusters. Partition reassignment was performed using kafka-reassign-partitions.sh. The detailed process is as follows:

a. To reassign the partitions for a topic in a Kafka cluster, we generated the proposed reassignment config JSON (this was performed for both the clusters).

```bash
kafka-reassign-partitions --bootstrap
-server=172.30.0.160:9092,172.30.0.172:9092,172.30.0.188:9092,172.30.0.123:9092
--broker-list "1,2,3,4" --topics-to-move-json-file /tmp/topics.json --generate
```

b. The generated reassignment JSON was then saved in /tmp/reassignment-file.json.

c. The actual partition reassignment process was triggered by the following command:

```bash
kafka-reassign-partitions --bootstrap
-server=172.30.0.198:9092,172.30.0.163:9092,172.30.0.221:9092,172.30.0.204:9092
--reassignment-json-file /tmp/reassignment-file.json --execute
```

7. After a few minutes when the reassignment was completed, another health check on the brokers showed that cluster using NFSv3 storage volumes had run into a silly rename issue and had crashed, whereas Cluster 1 using NetApp ONTAP NFSv4.1 storage volumes with the fix continued operations without any disruptions.

- Cluster1-Broker-1 is alive.
- Cluster2-broker-1 is dead.

8. Upon checking the Kafka log directories, it was clear that Cluster 1 using NetApp ONTAP NFSv4.1 storage volumes with the fix had clean partition assignment, while Cluster 2 using generic NFSv3 storage did not due to silly rename issues, which led to the crash. The following picture shows partition rebalancing of Cluster 2, which resulted in a silly rename issue on NFSv3 storage.
The following picture shows a clean partition rebalancing of Cluster 1 using NetApp NFSv4.1 storage.

Why NetApp NFS for Kafka workloads?

Now that there is a solution for the silly rename issue in NFS storage with Kafka, you can create robust deployments that leverage NetApp ONTAP storage for your Kafka workload. Not only does this significantly reduce operational overhead, it also brings the following benefits to your Kafka clusters:

- **Reduced CPU utilization on Kafka brokers.** Using disaggregated NetApp ONTAP storage separates disk I/O operations from the broker and thus reduces its CPU footprint.

- **Faster broker recovery-time.** Since disaggregated NetApp ONTAP storage is shared across Kafka broker nodes, a new compute instance can replace a bad broker at any point in a fraction of the time compared to conventional Kafka deployments without rebuilding the data.

- **Storage efficiency.** As the storage layer of the application is now provisioned through NetApp ONTAP, customers can avail all the benefits of storage efficiency that comes with ONTAP, such as in-line data compression, deduplication, and compaction.
These benefits were tested and validated in test cases that we discuss in detail in this section.

**Reduced CPU utilization on Kafka broker**

We discovered that overall CPU utilization is lower than its DAS counterpart when we ran similar workloads on two separate Kafka clusters that were identical in their technical specifications but differed in their storage technologies. Not only is the overall CPU utilization lower when Kafka cluster is using ONTAP storage, but the increase in the CPU utilization demonstrated a gentler gradient than in a DAS-based Kafka cluster.

**Architectural setup**

The following table shows the environmental configuration used to demonstrate reduced CPU utilization.

<table>
<thead>
<tr>
<th>Platform component</th>
<th>Environment configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kafka 3.2.3</td>
<td>• 3 x zookeepers – t2.small</td>
</tr>
<tr>
<td>Benchmarking tool: OpenMessaging</td>
<td>• 3 x broker servers – i3en.2xlarge</td>
</tr>
<tr>
<td></td>
<td>• 1 x Grafana – c5n.2xlarge</td>
</tr>
<tr>
<td></td>
<td>• 4 x Producer/Consumer — c5n.2xlarge</td>
</tr>
<tr>
<td>Operating system on all nodes</td>
<td>RHEL 8.7 or later</td>
</tr>
<tr>
<td>NetApp Cloud Volumes ONTAP instance</td>
<td>Single Node Instance – M5.2xLarge</td>
</tr>
</tbody>
</table>

**Benchmarking tool**

The benchmarking tool used in this test case is the OpenMessaging framework. OpenMessaging is vendor-neutral and language-independent; it provides industry guidelines for finance, e-commerce, IoT, and big-data; and it helps develop messaging and streaming applications across heterogeneous systems and platforms. The following figure depicts the interaction of OpenMessaging clients with a Kafka cluster.
- **Compute.** We used a three-node Kafka cluster with a three-node zookeeper ensemble running on dedicated servers. Each broker had two NFSv4.1 mount points to a single volume on the NetApp CVO instance through a dedicated LIF.

- **Monitoring.** We used two nodes for a Prometheus-Grafana combination. For generating workloads, we have a separate three-node cluster that can produce to and consume from this Kafka cluster.

- **Storage.** We used a single-node NetApp Cloud volumes ONTAP instance with six 250GB GP2 AWS-EBS volumes mounted on the instance. These volumes were then exposed to the Kafka cluster as six NFSv4.1 volumes through dedicated LIFs.

- **Configuration.** The two configurable elements in this test case were Kafka brokers and OpenMessaging workloads.
  - **Broker config.** The following specifications were selected for the Kafka brokers. We used replication factor of 3 for all measurements, as is highlighted below.
OpenMessaging benchmark (OMB) workload config. The following specifications were provided. We specified a target producer rate, highlighted below.

```
broker.id=1
advertised.listeners=PLAINTEXT://172.30.0.185:9092
log.dirs=/mnt/data-1
zookeeper.connect=172.30.0.13:2181,172.30.0.108:2181,172.30.0.253:2181
num.replica.fetchers=8
message.max.bytes=10485760
replica.fetch.max.bytes=10485760
num.network.threads=8

default.replication.factor=3
replica.lag.time.max.ms=100000000
replica.fetch.max.bytes=1048576
replica.fetch.wait.max.ms=500
num.replica.fetchers=1
replica.high.watermark.checkpoint.interval.ms=5000
fetch.purgatory.purge.interval.requests=1000
producer.purgatory.purge.interval.requests=1000
replica.socket.timeout.ms=30000
replica.socket.receive.buffer.bytes=65536
```

- • OpenMessaging benchmark (OMB) workload config. The following specifications were provided. We specified a target producer rate, highlighted below.

```
name: 4 producer / 4 consumers on 1 topic
topics: 1
partitionsPerTopic: 100
messageSize: 1024
payloadFile: "payload/payload-1Kb.data"
subscriptionsPerTopic: 1
consumerPerSubscription: 4
producersPerTopic: 4
producerRate: 40000
consumerBacklogSizeGB: 0
testDurationMinutes: 5
```

Methodology of testing

1. Two similar clusters were created, each having its own set of benchmarking cluster swarms.
   - • Cluster 1. NFS-based Kafka cluster.
   - • Cluster 2. DAS-based Kafka cluster.

2. Using an OpenMessaging command, similar workloads were triggered on each cluster.

```
sudo bin/benchmark --drivers driver-kafka/kafka-group-all.yaml
workloads/1-topic-100-partitions-1kb.yaml
```
3. The produce rate configuration was increased in four iterations, and CPU utilization was recorded with Grafana. The produce rate was set to the following levels:

- 10,000
- 40,000
- 80,000
- 100,000

**Observation**

There are two primary benefits of using NetApp NFS storage with Kafka:

- **You can reduce CPU usage by almost one-third.** The overall CPU usage under similar workloads was lower for NFS compared to DAS SSDs; savings range from 5% for lower produce rates to 32% for higher produce rates.

- **A three-fold reduction in CPU utilization drift at higher produce rates.** As expected, there was an upward drift for the increase in CPU utilization as the produce rates were increased. However, CPU utilization on Kafka brokers using DAS went up from 31% for the lower produce rate to 70% for the higher produce rate, a 39% increase. However, with an NFS storage backend, the CPU utilization went up from 26% to 38%, a 12% increase.
Also, at 100,000 messages, DAS shows more CPU utilization than an NFS cluster.
**Faster broker recovery**

We discovered that Kafka brokers recover faster when they are using shared NetApp NFS storage. When a broker crashes in a Kafka cluster, this broker can be replaced by a healthy broker with the same broker ID. Upon performing this test case, we found that, in the case of a DAS-based Kafka cluster, the cluster rebuilds the data on a newly added healthy broker, which is time consuming. In the case of a NetApp NFS-based Kafka cluster, the replacing broker continues to read data from the previous log directory and recovers much faster.

**Architectural setup**

The following table shows the environmental configuration for a Kafka cluster using NAS.

<table>
<thead>
<tr>
<th>Platform component</th>
<th>Environment configuration</th>
</tr>
</thead>
</table>
| Kafka 3.2.3                 | • 3 x zookeepers – t2.small  
                                | • 3 x broker servers – i3en.2xlarge            |
|                             | • 1 x Grafana – c5n.2xlarge                       |
|                             | • 4 x producer/consumer — c5n.2xlarge            |
|                             | • 1 x backup Kafka node – i3en.2xlarge           |
| Operating system on all nodes | RHEL8.7 or later                                |
| NetApp Cloud Volumes ONTAP instance | Single-node instance – M5.2xLarge                |

The following figure depicts the architecture of an NAS-based Kafka cluster.

![AWS Cloud Architecture Diagram](image-url)
• **Compute.** A three-node Kafka cluster with a three-node zookeeper ensemble running on dedicated servers. Each broker has two NFS mount points to a single volume on the NetApp CVO instance via a dedicated LIF.

• **Monitoring.** Two nodes for a Prometheus-Grafana combination. For generating workloads, we use a separate three-node cluster that can produce and consume to this Kafka cluster.

• **Storage.** A single-node NetApp Cloud volumes ONTAP instance with six 250GB GP2 AWS-EBS volumes mounted on the instance. These volumes are then exposed to the Kafka cluster as six NFS volume through dedicated LIFs.

• **Broker configuration.** The one configurable element in this test case are Kafka brokers. The following specifications were selected for the Kafka brokers. The `replica.lag.time.max.ms` is set to a high value because this determines how fast a particular node is taken out of the ISR list. When you switch between bad and healthy nodes, you don’t want that broker ID to be excluded from the ISR list.

```
broker.id=1
advertised.listeners=PLAINTEXT://172.30.0.108:9092
log.dirs=/mnt/data-1
zookeeper.connect=172.30.0.13:2181,172.30.0.108:2181,172.30.0.253:2181
num.replica.fetchers=8
message.max.bytes=10485760
replica.fetch.max.bytes=10485760
num.network.threads=8
default.replication.factor=3
replica.lag.time.max.ms=500000000
replica.fetch.max.bytes=1048576
replica.fetch.wait.max.ms=500
num.replica.fetchers=1
replica.high.watermark.checkpoint.interval.ms=5000
fetch.purgatory.purge.interval.requests=1000
producer.purgatory.purge.interval.requests=1000
replica.socket.timeout.ms=30000
replica.socket.receive.buffer.bytes=65536
```

**Methodology of testing**

1. Two similar clusters were created:
   - An EC2-based confluent cluster.
   - A NetApp NFS-based confluent cluster.

2. One standby Kafka node was created with a configuration identical to the nodes from the original Kafka cluster.

3. On each of the clusters, a sample topic was created, and approximately 110GB of data was populated on each of the brokers.

   - **EC2-based cluster.** A Kafka broker data directory is mapped on `/mnt/data-2` (In the following figure, Broker-1 of cluster1 [left terminal]).

   - **NetApp NFS-based cluster.** A Kafka broker data directory is mounted on NFS point `/mnt/data` (In the following figure, Broker-1 of cluster2 [right terminal]).
4. In each of the clusters, Broker-1 was terminated to trigger a failed broker recovery process.

5. After the broker was terminated, the broker IP address was assigned as a secondary IP to the standby broker. This was necessary because a broker in a Kafka cluster is identified by the following:
   - **IP address.** Assigned by reassigning the failed broker IP to the standby broker.
   - **Broker ID.** This was configured in the standby broker server.properties.

6. Upon IP assignment, the Kafka service was started on the standby broker.

7. After a while, the server logs were pulled to check the time taken to build data on the replacement node in the cluster.

**Observation**

Kafka broker recovery was almost nine times faster. The time it took to recover a failed broker node was found to be significantly faster when using NetApp NFS shared storage compared to using DAS SSDs in a Kafka cluster. For 1TB of topic data, the recovery time for a DAS-based cluster was 48 minutes, compared to less than 5 minutes for a NetApp-NFS based Kafka cluster.

We observed that the EC2-based cluster took 10 minutes to rebuild the 110GB of data on the new broker node, whereas the NFS-based cluster completed the recovery in 3 minutes. We also observed in the in logs that consumer offsets for the partitions for EC2 were 0, while, on the NFS cluster, consumer offsets were picked up from the previous broker.

```
```

**DAS-based cluster**

1. The backup node started at 08:55:53.730.

```
3 [2022-10-31 08:55:53,727] INFO Registered signal handlers for TERM, INT, HUP (or)
```

2. The data rebuilding process ended at 09:05:24,860. Processing 110GB of data required approximately 10 minutes.
NFS-based cluster

1. The backup node was started at 09:39:17,213. The starting log entry is highlighted below.

2. The data rebuild process ended at 09:42:29,115. Processing 110GB of data required approximately 3 minutes.

The test was repeated for brokers containing around 1TB data, which took approximately 48 minutes for the DAS and 3 min for NFS. The results are depicted in the following graph.
**Storage efficiency**

Because the storage layer of the Kafka cluster was provisioned through NetApp ONTAP, we got all the storage efficiency capabilities of ONTAP. This was tested by generating a significant amount of data on a Kafka cluster with NFS storage provisioned on Cloud Volumes ONTAP. We could see that there was a significant space reduction due to ONTAP capabilities.

**Architectural setup**

The following table shows the environmental configuration for a Kafka cluster using NAS.

<table>
<thead>
<tr>
<th>Platform component</th>
<th>Environment configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kafka 3.2.3</td>
<td>• 3 x zookeepers – t2.small&lt;br&gt;• 3 x broker servers – i3en.2xlarge&lt;br&gt;• 1 x Grafana – c5n.2xlarge&lt;br&gt;• 4 x producer/consumer — c5n.2xlarge</td>
</tr>
<tr>
<td>Operating system on all nodes</td>
<td>RHEL8.7 or later</td>
</tr>
<tr>
<td>NetApp Cloud Volumes ONTAP instance</td>
<td>Single node instance – M5.2xLarge</td>
</tr>
</tbody>
</table>

The following figure depicts the architecture of an NAS-based Kafka cluster.

- **Compute.** We used a three-node Kafka cluster with a three-node zookeeper ensemble running on dedicated servers. Each broker had two NFS mount points to a single volume on the NetApp CVO instance
via a dedicated LIF.

- **Monitoring.** We used two nodes for a Prometheus-Grafana combination. For generating workloads, we used a separate three-node cluster that could produce and consume to this Kafka cluster.

- **Storage.** We used a single-node NetApp Cloud Volumes ONTAP instance with six 250GB GP2 AWS-EBS volumes mounted on the instance. These volumes were then exposed to the Kafka cluster as six NFS volumes through dedicated LIFs.

- **Configuration.** The configurable elements in this test case were the Kafka brokers.

Compression was switched off on the producer’s end, thus enabling producers to generate high throughput. Storage efficiency was instead handled by the compute layer.

**Methodology of testing**

1. A Kafka cluster was provisioned with the specifications mentioned above.
2. On the cluster, about 350GB data was produced using the OpenMessaging Benchmarking tool.
3. After the workload was completed, the storage efficiency statistics were collected using ONTAP System Manager and the CLI.

**Observation**

For data that was generated using the OMB tool, we saw space savings of ~33% with a storage efficiency ratio of 1.70:1. As seen in the following figures, the logical space used by the data produced was 420.3GB and the physical space used to hold the data was 281.7GB.
1.7 to 1 Data Reduction
420 GiB logical used

IOPS: 3  |  Latency: 1.00 ms
Throughput: 0.22 MB/s

Name of the Aggregate: aggr1
Node where Aggregate Resides: shantanuCVOinstancenew-01
Total Storage Efficiency Ratio: 1.70:1
Total Data Reduction Efficiency Ratio Without Snapshots: 1.70:1
Total Data Reduction Efficiency Ratio without snapshots and flexclones: 1.70:1
Logical Space Used for All Volumes: 420.3GB
Physical Space Used for All Volumes: 281.7GB
Performance overview and validation in AWS

A Kafka cluster with the storage layer mounted on NetApp NFS was benchmarked for performance in the AWS cloud. The benchmarking examples are described in the following sections.

Kafka in AWS cloud with NetApp Cloud Volumes ONTAP (high-availability pair and single node)

A Kafka cluster with NetApp Cloud Volumes ONTAP (HA pair) was benchmarked for performance in the AWS cloud. This benchmarking is described in the following sections.

Architectural setup

The following table shows the environmental configuration for a Kafka cluster using NAS.

<table>
<thead>
<tr>
<th>Platform component</th>
<th>Environment configuration</th>
</tr>
</thead>
</table>
| Kafka 3.2.3        | • 3 x zookeepers – t2.small  
|                    | • 3 x broker servers – i3en.2xlarge  
|                    | • 1 x Grafana – c5n.2xlarge  
|                    | • 4 x producer/consumer — c5n.2xlarge  |

Operating system on all nodes RHEL8.6

NetApp Cloud Volumes ONTAP instance

| HA pair instance – m5dn.12xLarge x 2node  
| Single Node Instance - m5dn.12xLarge x 1 node  |

NetApp cluster volume ONTAP setup

1. For the Cloud Volumes ONTAP HA pair, we created two aggregates with three volumes on each aggregate on each storage controller. For the single Cloud Volumes ONTAP node, we create six volumes in an aggregate.
2. To achieve better network performance, we enabled high speed networking for both the HA pair and the single node.
3. We noticed that the ONTAP NVRAM had more IOPS so we changed the IOPS to 2350 for the Cloud Volumes ONTAP root volume. The root volume disk in Cloud Volumes ONTAP was 47GB in size. The following ONTAP command is for the HA pair, and the same step is applicable for the single node.
statistics start -object vnvram -instance vnvram -counter backing_store_iops -sample-id sample_555
kafka_nfs_cvo_ha1::*> statistics show -sample-id sample_555
Object: vnvram
Instance: vnvram
Start-time: 1/18/2023 18:03:11
End-time: 1/18/2023 18:03:13
Elapsed-time: 2s
Scope: kafka_nfs_cvo_ha1-01

<table>
<thead>
<tr>
<th>Counter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>backing_store_iops</td>
<td>1479</td>
</tr>
</tbody>
</table>

Object: vnvram
Instance: vnvram
Start-time: 1/18/2023 18:03:11
End-time: 1/18/2023 18:03:13
Elapsed-time: 2s
Scope: kafka_nfs_cvo_ha1-02

<table>
<thead>
<tr>
<th>Counter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>backing_store_iops</td>
<td>1210</td>
</tr>
</tbody>
</table>

2 entries were displayed.
kafka_nfs_cvo_ha1::*>
The following figure depicts the architecture of an NAS-based Kafka cluster.

- **Compute.** We used a three-node Kafka cluster with a three-node zookeeper ensemble running on dedicated servers. Each broker had two NFS mount points to a single volume on the Cloud Volumes ONTAP instance through a dedicated LIF.
- **Monitoring.** We used two nodes for a Prometheus-Grafana combination. For generating workloads, we used a separate three-node cluster that could produce and consume to this Kafka cluster.
- **Storage.** We used an HA-pair Cloud volumes ONTAP instance with one 6TB GP3 AWS-EBS volume mounted on the instance. The volume was then exported to the Kafka broker with an NFS mount.
OpenMessage Benchmarking configurations

1. For better NFS performance, we need more network connections between the NFS server and the NFS client, which can be created using nconnect. Mount the NFS volumes on the broker nodes with the nconnect option by running the following command:
2. Check the network connections in Cloud Volumes ONTAP. The following ONTAP command is used from the single Cloud Volumes ONTAP node. The same step is applicable to the Cloud Volumes ONTAP HA pair.

```
Last login time: 1/20/2023 00:16:29
kafka_nfs_cvo_sn::> network connections active show -service nfs* -fields remote-host
```

3. We use the following Kafka server.properties in all Kafka brokers for the Cloud Volumes ONTAP HA pair. The log.dirs property is different for each broker, and the remaining properties are common for brokers. For broker1, the log.dirs value is as follows:

```bash
[root@ip-172-30-0-121 ~]# cat /opt/kafka/config/server.properties
broker.id=0
advertised.listeners=PLAINTEXT://172.30.0.121:9092
log.dirs=/kafka_aggr3_vol1/broker1,/kafka_aggr3_vol2/broker1,/kafka_aggr3_vol3/broker1,/kafka_aggr22_vol1/broker1,/kafka_aggr22_vol2/broker1,/kafka_aggr22_vol3/broker1
zookeeper.connect=172.30.0.12:2181,172.30.0.30:2181,172.30.0.178:2181
num.network.threads=64
num.io.threads=64
socket.send.buffer.bytes=102400
socket.receive.buffer.bytes=102400
socket.request.max.bytes=104857600
num.partitions=1
num.recovery.threads.per.data.dir=1
offsets.topic.replication.factor=1
transaction.state.log.replication.factor=1
transaction.state.log.min.isr=1
replica.fetch.max.bytes=524288000
background.threads=20
num.replica.alter.log.dirs.threads=40
num.replica.fetchers=20
[root@ip-172-30-0-121 ~]#
```

- For broker2, the log.dirs property value is as follows:

```bash
log.dirs=/kafka_aggr3_vol1/broker2,/kafka_aggr3_vol2/broker2,/kafka_aggr3_vol3/broker2,/kafka_aggr22_vol1/broker2,/kafka_aggr22_vol2/broker2,/kafka_aggr22_vol3/broker2
```

- For broker3, the log.dirs property value is as follows:
log.dirs=/kafka_aggr3_vol1/broker3,/kafka_aggr3_vol2/broker3,/kafka_aggr3_vol3/broker3,/kafka_aggr22_vol1/broker3,/kafka_aggr22_vol2/broker3,/kafka_aggr22_vol3/broker3

4. For the single Cloud Volumes ONTAP node, The Kafka `servers.properties` is the same as for the Cloud Volumes ONTAP HA pair except for the `log.dirs` property.
   - For broker1, the `log.dirs` value is as follows:

```bash
```

   - For broker2, the `log.dirs` value is as follows:

```bash
```

   - For broker3, the `log.dirs` property value is as follows:

```bash
```

5. The workload in the OMB is configured with the following properties: (/opt/benchmark/workloads/1-topic-100-partitions-1kb.yaml).

```yaml
topics: 4
partitionsPerTopic: 100
messageSize: 32768
useRandomizedPayloads: true
randomBytesRatio: 0.5
randomizedPayloadPoolSize: 100
subscriptionsPerTopic: 1
consumerPerSubscription: 80
producersPerTopic: 40
producerRate: 1000000
consumerBacklogSizeGB: 0
testDurationMinutes: 5
```

The `messageSize` can vary for each use case. In our performance test, we used 3K.
We used two different drivers, Sync or Throughput, from OMB to generate the workload on the Kafka cluster.

- The yaml file used for Sync driver properties is as follows (/opt/benchmark/driver-kafka/kafka-sync.yaml):

```yaml
name: Kafka
driverClass:
io.openmessaging.benchmark.driver.kafka.KafkaBenchmarkDriver # Kafka client-specific configuration
replicationFactor: 3
topicConfig: |
  min.insync.replicas=2
  flush.messages=1
  flush.ms=0
commonConfig: |

bootstrap.servers=172.30.0.121:9092,172.30.0.72:9092,172.30.0.223:9092
producerConfig: |
  acks=all
  linger.ms=1
  batch.size=1048576
consumerConfig: |
  auto.offset.reset=earliest
  enable.auto.commit=false
  max.partition.fetch.bytes=10485760
```

- The yaml file used for the Throughput driver properties is as follows (/opt/benchmark/driver-kafka/kafka-throughput.yaml):
Methodology of testing

1. A Kafka cluster was provisioned as per the specification described above using Terraform and Ansible. Terraform is used to build the infrastructure using AWS instances for the Kafka cluster and Ansible builds the Kafka cluster on them.

2. An OMB workload was triggered with the workload configuration described above and the Sync driver.

Sudo bin/benchmark -drivers driver-kafka/kafka-sync.yaml workloads/1-topic-100-partitions-1kb.yaml

3. Another workload was triggered with the Throughput driver with same workload configuration.

sudo bin/benchmark -drivers driver-kafka/kafka-throughput.yaml workloads/1-topic-100-partitions-1kb.yaml

Observation

Two different types of drivers were used to generate workloads to benchmark the performance of a Kafka instance running on NFS. The difference between the drivers is the log flush property.

For a Cloud Volumes ONTAP HA pair:
• Total throughput generated consistently by the Sync driver: ~1236 MBps.
• Total throughput generated for the Throughput driver: peak ~1412 MBps.

For a single Cloud Volumes ONTAP node:

• Total throughput generated consistently by the Sync driver: ~1962MBps.
• Total throughput generated by the Throughput driver: peak ~1660MBps

The Sync driver can generate consistent throughput as logs are flushed to the disk instantly, whereas the Throughput driver generates bursts of throughput as logs are committed to disk in bulk.

These throughput numbers are generated for the given AWS configuration. For higher performance requirements, the instance types can be scaled up and tuned further for better throughput numbers. The total throughput or total rate is the combination of both producer and consumer rate.

Be sure to check the storage throughput when performing throughput or sync driver benchmarking.
A Kafka cluster with the storage layer mounted on NetApp NFS was benchmarked for performance in the AWS FSx for NetApp ONTAP. The benchmarking examples are described in the following sections.
Apache Kafka in AWS FSx for NetApp ONTAP

Network File System (NFS) is a widely used network filesystem for storing large amounts of data. In most organizations data is increasingly being generated by streaming applications like Apache Kafka. These workloads require scalability, low latency, and a robust data ingestion architecture with modern storage capabilities. To enable real-time analytics and to provide actionable insights, a well designed and highly performant infrastructure is required.

Kafka by design works with POSIX compliant file system and relies on the file system to handle file operations, but when storing data on an NFSv3 file system, the Kafka broker NFS client can interpret file operations differently from a local file system like XFS or Ext4. A common example is the NFS Silly rename which caused Kafka brokers to fail when expanding clusters and re-allocating partitions. To deal with this challenge NetApp has updated the open-source Linux NFS client with changes now generally available in RHEL8.7, RHEL9.1, and supported from the current FSx for NetApp ONTAP release, ONTAP 9.12.1.

Amazon FSx for NetApp ONTAP provides a fully managed, scalable, and highly performance NFS file system in the cloud. Kafka data on FSx for NetApp ONTAP can scale to handle large amounts of data and ensure fault tolerance. NFS provides centralized storage management and data protection for critical and sensitive datasets.

These enhancements make it possible for AWS customer to take advantage of FSx for NetApp ONTAP when running Kafka workloads on AWS compute services. These benefits are:

- Reducing CPU utilization to reduce the I/O wait time
- Faster Kafka broker recovery time.
- Reliability and efficiency.
- Scalability and performance.
- Multi-Availability Zone availability.
- Data protection.

Performance overview and validation in AWS FSx for NetApp ONTAP

A Kafka cluster with the storage layer mounted on NetApp NFS was benchmarked for performance in the AWS cloud. The benchmarking examples are described in the following sections.

Kafka in AWS FSx for NetApp ONTAP

A Kafka cluster with AWS FSx for NetApp ONTAP was benchmarked for performance in the AWS cloud. This benchmarking is described in the following sections.

Architectural setup

The following table shows the environmental configuration for a Kafka cluster using AWS FSx for NetApp ONTAP.

<table>
<thead>
<tr>
<th>Platform component</th>
<th>Environment configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kafka 3.2.3</td>
<td>• 3 x zookeepers – t2.small</td>
</tr>
<tr>
<td></td>
<td>• 3 x broker servers – i3en.2xlarge</td>
</tr>
<tr>
<td></td>
<td>• 1 x Grafana – c5n.2xlarge</td>
</tr>
<tr>
<td></td>
<td>• 4 x producer/consumer — c5n.2xlarge</td>
</tr>
<tr>
<td>Operating system on all nodes</td>
<td>RHEL8.6</td>
</tr>
</tbody>
</table>
### NetApp FSx for NetApp ONTAP setup

1. For our initial testing, we have created a FSx for NetApp ONTAP filesystem with 2TB of capacity and 40000 IOPs for 2GB/Sec throughput.

   ```
   [root@ip-172-31-33-69 ~]# aws fsx create-file-system --region us-east-2
   --storage-capacity 2048 --subnet-ids <desired subnet 1> subnet-<desired subnet 2> --file-system-type ONTAP --ontap-configuration
   DeploymentType=MULTI_AZ_HA_1,ThroughputCapacity=2048,PreferredSubnetId=<desired primary subnet>,FsxAdminPassword=<new password>,DiskIopsConfiguration="{Mode=USER_PROVISIONED,Iops=40000}"
   ```

   In our example, we are deploying FSx for NetApp ONTAP through the AWS CLI. You will need to customize the command further in your environment as needed. FSx for NetApp ONTAP can additionally be deployed and managed through the AWS Console for an easier and more streamlined deployment experience with less command line input.

   Documentation In FSx for NetApp ONTAP, the max IOPS achievable for a 2GB/Sec throughput filesystem in our test region (US-East-1) is 80,000 iops. The total max iops for a FSx for NetApp ONTAP filesystem is 160,000 iops which requires a 4GB/Sec throughput deployment to achieve which we will demonstrate later in this document.

   For more information on FSx for NetApp ONTAP performance specifications, please feel free to visit the AWS FSx for NetApp ONTAP documentation here: [https://docs.aws.amazon.com/fsx/latest/ONTAPGuide/performance.html](https://docs.aws.amazon.com/fsx/latest/ONTAPGuide/performance.html).

   Detailed command line syntax for FSx “create-file-system” can be found here: [https://docs.aws.amazon.com/cli/latest/reference/fsx/create-file-system.html](https://docs.aws.amazon.com/cli/latest/reference/fsx/create-file-system.html)

   For instance, you can specify a specific KMS key as opposed to the default AWS FSx master key that is used when no KMS key is specified.

2. While creating the FSx for NetApp ONTAP filesystem, Wait till the “LifeCycle” status changes to “AVAILABLE” in your JSON return after describing your filesystem as follows:

   ```
   [root@ip-172-31-33-69 ~]# aws fsx describe-file-systems --region us-east-1 --file-system-ids fs-02ff04bab5ce01c7c
   ```

3. Validate the credentials by login into Fsx for NetApp ONTAP SSH with the fsxadmin user:
   Fsxadmin is the default admin account for FSx for NetApp ONTAP filesystems at creation. The password for fsxadmin is the password that was configured when first creating the filesystem either in the AWS Console or with the AWS CLI as we completed in Step 1.
4. Once your credentials have been validated, Create the storage Virtual Machine on the FSx for NetApp ONTAP filesystem

```bash
[root@ip-172-31-33-69 ~]# aws fsx --region us-east-1 create-storage-virtual-machine --name svmkafkatest --file-system-id fs-02ff04bab5ce01c7c
```

A Storage Virtual Machine (SVM) is an isolated file server with its own administrative credentials and endpoints for administering and accessing data in FSx for NetApp ONTAP volumes and provides FSx for NetApp ONTAP multi-tenancy.

5. Once you have configured your primary Storage Virtual Machine, SSH into the newly created FSx for NetApp ONTAP filesystem and create volumes in storage virtual machine using below sample command and similarly we create 6 volumes for this validation. Based on our validation, keep the default constituent (8) or less constituents which will provides better performance to kafka.

```bash
FsxId02ff04bab5ce01c7c::*> volume create -volume kafkafsxN1 -state online -policy default -unix-permissions ---rwxr-xr-x -junction-active true -type RW -snapshot-policy none -junction-path /kafkafsxN1 -aggr -list aggr1
```

6. We will need additional capacity in our volumes for our testing. Extend the size of the volume to 2TB and mount on the junction path.

```bash
FsxId02ff04bab5ce01c7c::*> volume size -volume kafkafsxN1 -new-size +2TB
vol size: Volume "svmkafkatest:kafkafsxN1" size set to 2.10t.

FsxId02ff04bab5ce01c7c::*> volume size -volume kafkafsxN2 -new-size +2TB
vol size: Volume "svmkafkatest:kafkafsxN2" size set to 2.10t.

FsxId02ff04bab5ce01c7c::*> volume size -volume kafkafsxN3 -new-size +2TB
```
vol size: Volume "svmkafkatest:kafkafsxN3" size set to 2.10t.

FsxFd02ff04bab5ce01c7c::*> volume size -volume kafkafsxN4 -new-size +2TB
vol size: Volume "svmkafkatest:kafkafsxN4" size set to 2.10t.

FsxFd02ff04bab5ce01c7c::*> volume size -volume kafkafsxN5 -new-size +2TB
vol size: Volume "svmkafkatest:kafkafsxN5" size set to 2.10t.

FsxFd02ff04bab5ce01c7c::*> volume size -volume kafkafsxN6 -new-size +2TB
vol size: Volume "svmkafkatest:kafkafsxN6" size set to 2.10t.

FsxFd02ff04bab5ce01c7c::*> volume mount -volume kafkafsxN1 -junction -path /kafkafsxN1
FsxFd02ff04bab5ce01c7c::*> volume mount -volume kafkafsxN2 -junction -path /kafkafsxN2
In FSx for NetApp ONTAP, volumes can be thin provisioned. In our example, the total extended volume capacity exceeds total filesystem capacity so we will need to extend the total filesystem capacity in order to unlock additional provisioned volume capacity which we will demonstrate in our next step.

7. Next, for additional performance and capacity, we extend the FSx for NetApp ONTAP throughput capacity from 2GB/Sec to 4GB/Sec and IOPS to 160000, and capacity to 5 TB

```
[root@ip-172-31-33-69 ~]# aws fsx update-file-system --region us-east-1 --storage-capacity 5120 --ontap-configuration 'ThroughputCapacity=4096,DiskIopsConfiguration={Mode=USER_PROVISIONED,Iops=160000}' --file-system-id fs-02ff04bab5ce01c7c
```

Detailed command line syntax for FSx "update-file-system" can be found here:

8. The FSx for NetApp ONTAP volumes are mounted with nconnect and default opions in Kafka brokers

The following picture shows our final architecture of a our FSx for NetApp ONTAP based Kafka cluster:
\[\text{Compute. We used a three-node Kafka cluster with a three-node zookeeper ensemble running on dedicated servers. Each broker had six NFS mount points to a six volumes on the FSx for NetApp ONTAP instance.}\]

\[\text{Monitoring. We used two nodes for a Prometheus-Grafana combination. For generating workloads, we used a separate three-node cluster that could produce and consume to this Kafka cluster.}\]

\[\text{Storage. We used an FSx for NetApp ONTAP with six 2TB volumes mounted. The volume was then exported to the Kafka broker with an NFS mount. The FSx for NetApp ONTAP volumes are mounted with 16 nconnect sessions and default options in Kafka brokers.}\]

**OpenMessage Benchmarking configurations.**

We used the same configuration used for the NetApp Cloud volumes ONTAP and their details are here - https://docs.netapp.com/us-en/netapp-solutions/data-analytics/kafka-nfs-performance-overview-and-validation-in-aws.html#architectural-setup

**Methodology of testing**

1. A Kafka cluster was provisioned as per the specification described above using terraform and ansible. Terraform is used to build the infrastructure using AWS instances for the Kafka cluster and ansible builds the Kafka cluster on them.

2. An OMB workload was triggered with the workload configuration described above and the Sync driver.

\[
\text{sudo bin/benchmark -drivers driver-kafka/kafka-sync.yaml workloads/1-topic-100-partitions-1kb.yaml}
\]

3. Another workload was triggered with the Throughput driver with same workload configuration.
Observation

Two different types of drivers were used to generate workloads to benchmark the performance of a Kafka instance running on NFS. The difference between the drivers is the log flush property.

For a Kafka Replication factor 1 and the FSx for NetApp ONTAP:

- Total throughput generated consistently by the Sync driver: ~ 3218 MBps and peak performance in ~ 3652 MBps.
- Total throughput generated consistently by the Throughput driver: ~ 3679 MBps and peak performance in ~ 3908 MBps.

For Kafka with replication factor 3 and the FSx for NetApp ONTAP:

- Total throughput generated consistently by the Sync driver: ~ 1252 MBps and peak performance in ~ 1382 MBps.
- Total throughput generated consistently by the Throughput driver: ~ 1218 MBps and peak performance in ~ 1328 MBps.

In Kafka replication factor 3, the read and write operation happened three times on the FSx for NetApp ONTAP.
In Kafka replication factor 1, the read and write operation is one time on the FSx for NetApp ONTAP, so in both validation, we able to reach the maximum throughput of 4GB/Sec.

The Sync driver can generate consistent throughput as logs are flushed to the disk instantly, whereas the Throughput driver generates bursts of throughput as logs are committed to disk in bulk.

These throughput numbers are generated for the given AWS configuration. For higher performance requirements, the instance types can be scaled up and tuned further for better throughput numbers. The total throughput or total rate is the combination of both producer and consumer rate.

The below chart shows the 2GB/Sec FSx for NetApp ONTAP and 4GB/Sec performance for Kafka replication factor 3. The replication factor 3 does the read and write operation three times on the FSx for NetApp ONTAP storage. The total rate for throughput driver is 881 MB/Sec, which does read and write Kafka operation approximately 2.64 GB/Sec on the 2GB/Sec FSx for NetApp ONTAP filesystem and total rate for throughput driver is 1328 MB/Sec that does read and write kafka operation approximately 3.98 GB/Sec. Ther Kafka
Performance is linear and scalable based on the FSx for NetApp ONTAP throughput.

The below chart shows the performance between EC2 instance vs FSx for NetApp ONTAP (Kafka Replication Factor: 3)

Performance overview and validation with AFF A900 on-premises

On-premises, we used the NetApp AFF A900 storage controller with ONTAP 9.12.1RC1 to validate the performance and scaling of a Kafka cluster. We used the same testbed as in our previous tiered storage best practices with ONTAP and AFF.

We used Confluent Kafka 6.2.0 to evaluate the AFF A900. The cluster features eight broker nodes and three zookeeper nodes. For performance testing, we used five OMB worker nodes.
Storage configuration

We used NetApp FlexGroups instances to provide a single namespace for log directories, simplifying recovery and configuration. We used NFSv4.1 and pNFS to provide direct path access to log segment data.

Client tuning

Each client mounted the FlexGroup instance with the following command.

```bash
mount -t nfs -o vers=4.1,nconnect=16 172.30.0.121:/kafka_vol01 /data/kafka_vol01
```

In addition, we increased the `max_session_slots` from the default 64 to 180. This matches the default session slot limit in ONTAP.

Kafka broker tuning

To maximize throughput in the system under test, we significantly increased the default parameters for certain key thread pools. We recommend following Confluent Kafka best practices for most configurations. This tuning was used to maximize the concurrency of outstanding I/O to storage. These parameters can be adjusted to match your broker’s compute resources and storage attributes.

```bash
num.io.threads=96
num.network.threads=96
background.threads=20
num.replica.alter.log.dirs.threads=40
num.replica.fetchers=20
queued.max.requests=2000
```
Workload generator testing methodology

We used the same OMB configurations as for cloud testing for the Throughput driver and topic configuration.

1. A FlexGroup instance was provisioned using Ansible on an AFF cluster.

```yaml
---
- name: Set up kafka broker processes
  hosts: localhost
  vars:
    ntap_hostname: 'hostname'
    ntap_username: 'user'
    ntap_password: 'password'
    size: 10
    size_unit: tb
    vserver: vs1
    state: present
    https: true
    export_policy: default
  volumes:
    - name: kafka_fg_vol01
      aggr: ["aggr1_a", "aggr2_a", "aggr1_b", "aggr2_b"]
      path: /kafka_fg_vol01
  tasks:
    - name: Edit volumes
      netapp.ontap.na_ontap_volume:
        state: "{{ state }}"
        name: "{{ item.name }}"
        aggr_list: "{{ item.aggr }}"
        aggr_list_multiplier: 8
        size: "{{ size }}"
        size_unit: "{{ size_unit }}"
        vserver: "{{ vserver }}"
        snapshot_policy: none
        export_policy: default
        junction_path: "{{ item.path }}"
        qos_policy_group: none
        wait_for_completion: True
        hostname: "{{ ntap_hostname }}"
        username: "{{ ntap_username }}"
        password: "{{ ntap_password }}"
        https: "{{ https }}"
        validate_certs: false
        connection: local
        with_items: "{{ volumes }}"
2. pNFS was enabled on the ONTAP SVM.

```
vserver modify -vserver vs1 -v4.1-pnfs enabled -tcp-max-xfer-size 262144
```

3. The workload was triggered with the Throughput driver using with same workload configuration as for Cloud Volumes ONTAP. See the section “Steady state performance” below. The workload used a replication factor of 3, meaning three copies of log segments were maintained in NFS.

```
sudo bin/benchmark --drivers driver-kafka/kafka-throughput.yaml workloads/1-topic-100-partitions-1kb.yaml
```

4. Finally, we completed measurements using a backlog to measure the ability of consumers to catch up to the latest messages. OMB constructs a backlog by pausing consumers during the beginning of a measurement. This produces three distinct phases: backlog creation (producer-only traffic), backlog draining (a consumer-heavy phase in which consumers catch up on missed events in a topic), and the steady state. See the section “exploring storage limits” for more information.

**Steady state performance**

We evaluated the AFF A900 using the OpenMessaging Benchmark to provide a similar comparison as for Cloud Volumes ONTAP in AWS and DAS in AWS. All performance values represent Kafka-cluster throughput at the producer and consumer level.

Steady state performance with Confluent Kafka and the AFF A900 achieved over 3.4GBps average throughput for both producer and consumers. This is over 3.4 million messages across the Kafka cluster. By visualizing the sustained throughput in bytes per second for BrokerTopicMetrics, we see the excellent steady state performance and traffic supported by the AFF A900.

This aligns well with the view of messages delivered per topic. The following graph provides a per-topic breakdown. In the configuration tested we saw nearly 900k messages per topic across four topics.
Extreme performance and exploring storage limits

For AFF, we also tested with OMB using the backlog feature. The backlog feature pauses consumer subscriptions while a backlog of events is built up in the Kafka cluster. During this phase, only producer traffic occurs, which generates events that are committed to logs. This most closely emulates batch processing or offline analytics workflows; in these workflows, consumer subscriptions are started and must read historical data that has already been evicted from the broker cache.

To understand the storage limitations on consumer throughput in this configuration, we measured the producer-only phase to understand how much write traffic the A900 could absorb. See the next section “Sizing guidance” to understand how to leverage this data.

During the producer-only part of this measurement, we saw high peak throughput that pushed the limits of A900 performance (when other broker resources were not saturated serving producer and consumer traffic).
We increased the message size to 16k for this measurement to limit per-message overheads and maximize storage throughput to NFS mount points.

messageSize: 16384  
consumerBacklogSizeGB: 4096

The Confluent Kafka cluster achieved a peak producer throughput of 4.03GBps.

18:12:23.833 [main] INFO WorkloadGenerator - Pub rate 257759.2 msg/s / 4027.5 MB/s | Pub err 0.0 err/s …

After OMB completed populating the event backlog, consumer traffic was restarted. During measurements with backlog draining, we observed peak consumer throughput of over 20GBps across all topics. The combined throughput to the NFS volume storing the OMB log data approached ~30GBps.

Sizing guidance

Amazon Web Services offers a sizing guide for Kafka cluster sizing and scaling.

This sizing provides a useful formula for determining storage throughput requirements for your Kafka cluster:

For an aggregated throughput produced into the cluster of $t_{cluster}$ with a replication factor of $r$, the throughput received by the broker storage is as follows:

$$t_{storage} = \frac{t_{cluster}}{\#\text{brokers}} + \frac{t_{cluster}}{\#\text{brokers}} \times (r - 1)$$

$$= \frac{t_{cluster}}{\#\text{brokers}} \times r$$

This can be simplified even further:

$$\max(t_{cluster}) \leq \max(t_{storage}) \times \frac{\#\text{brokers}}{r}$$

Using this formula allows you to select the appropriate ONTAP platform for your Kafka hot tier needs.

The following table explains the anticipated producer throughput for the A900 with different replication factors:

<table>
<thead>
<tr>
<th>Replication factor</th>
<th>Producer throughput (GPps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 (measured)</td>
<td>3.4</td>
</tr>
<tr>
<td>2</td>
<td>5.1</td>
</tr>
<tr>
<td>1</td>
<td>10.2</td>
</tr>
</tbody>
</table>

Conclusion

The NetApp solution for the silly rename problem provides a simple, inexpensive, and
centrally managed form of storage for workloads that were previously incompatible with NFS.

This new paradigm enables customers to create more manageable Kafka clusters that are easier to migrate and mirror for the purpose of disaster recovery and data protection. We have also seen that NFS provides additional benefits such as reduced CPU utilization and a faster recovery time, dramatically improved storage efficiency, and better performance through NetApp ONTAP.

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

- What is Apache Kafka?
  https://www.confluent.io/what-is-apache-kafka/

- What is silly rename?

- ONATP is read for streaming applications.
  https://www.netapp.com/blog/ontap-ready-for-streaming-applications/

- Silly-rename issue with Kafka.
  https://sbg.technology/2018/07/10/kafka-nfs/

- NetApp product documentation
  https://www.netapp.com/support-and-training/documentation/

- What is NFS?
  https://en.wikipedia.org/wiki/Network_File_System

- What is Kafka partition reassignment?
  https://docs.cloudera.com/runtime/7.2.10/kafka-managing/topics/kafka-manage-cli-reassign-overview.html

- What is the OpenMessaging Benchmark?
  https://openmessaging.cloud/

- How do you migrate a Kafka broker?

- How do you monitor Kafka broker with Prometheus?

- Managed Platform for Apache Kafka
Confluent Kafka with NetApp ONTAP storage controllers

TR-4941: Confluent with NetApp ONTAP storage controllers

Karthikeyan Nagalingam, Joe Scott, NetApp
Rankesh Kumar, Confluent

To make the Confluent Platform more scalable and elastic, it must be able to scale and balance workloads very quickly. Tiered storage makes storing huge volumes of data in Confluent manageable by reducing this operational burden.

The fundamental idea is to separate data storage from data processing, which makes it much easier to scale each independently.

Loaded with industry-leading innovations, NetApp ONTAP data management software provides Confluent with many advantages anywhere the data lives.

This document outlines performance benchmarks for the Confluent platform on NetApp ONTAP using a tiered storage benchmarking kit.

Solution

Confluent and NetApp AFF A900 storage controller powered by ONTAP are distributed systems designed for data streams. Both are horizontally scalable, fault tolerant, and provide excellent performance under load. They complement each other in distributed data streaming and stream processing with lower storage costs with data reduction technologies that minimize the data footprint. The AFF A900 storage controller provides great performance, while allowing the decoupling of compute and data storage resources. This simplifies system administration and allows resources to be scaled independently.
Solution architecture details

This section covers the hardware and software used for performance verification in Confluent Platform deployment with NetApp ONTAP for tiered storage. The following table covers the solution architecture and base components.

<table>
<thead>
<tr>
<th>Platform component</th>
<th>Environment configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confluent Platform version 6.2</td>
<td>• 3 x zookeepers</td>
</tr>
<tr>
<td></td>
<td>• 8 x broker servers</td>
</tr>
<tr>
<td></td>
<td>• 5 x tools servers</td>
</tr>
<tr>
<td></td>
<td>• 1 x Grafana</td>
</tr>
<tr>
<td></td>
<td>• 1 x control center</td>
</tr>
<tr>
<td>Operating system on all nodes</td>
<td>Linux (ubuntu 18.04)</td>
</tr>
<tr>
<td>NetApp ONTAP for warm buckets</td>
<td>• 1 x AFF A900 high-availability (HA) pair</td>
</tr>
<tr>
<td></td>
<td>• 4 x 24 x 800 SSDs</td>
</tr>
<tr>
<td></td>
<td>• S3 protocol</td>
</tr>
<tr>
<td></td>
<td>• 100GbE</td>
</tr>
<tr>
<td>15 Fujitsu PRIMERGY RX2540 servers</td>
<td>• 2 CPUs; 16 physical cores total</td>
</tr>
<tr>
<td></td>
<td>• Intel Xeon</td>
</tr>
<tr>
<td></td>
<td>• 256GB physical memory</td>
</tr>
<tr>
<td></td>
<td>• 100GbE dual port</td>
</tr>
</tbody>
</table>
**Technology overview**

This section describes the technology used in this solution.

**NetApp ONTAP storage controller**

NetApp ONTAP is a high-performance, enterprise-grade storage operating system.

NetApp ONTAP 9.8 introduces support for Amazon Simple Storage Service (S3) APIs. ONTAP supports a subset of Amazon Web Services (AWS) S3 API actions and allows data to be represented as objects in ONTAP-based systems across cloud providers (AWS, Azure, and GCP) and on-premises.

NetApp StorageGRID software is the flagship NetApp solution for object storage. ONTAP complements StorageGRID by providing an ingest and preprocessing point on the edge, expanding the data fabric powered by NetApp for object data, and increasing the value of the NetApp product portfolio.

Access to an S3 bucket is provided through authorized user and client applications. The following diagram shows the application accessing an S3 bucket.

![Diagram of S3 bucket access](image)

**Primary use cases**

The primary purpose of supporting S3 APIs is to provide objects access on ONTAP. The ONTAP unified storage architecture now supports files (NFS and SMB), blocks (FC and iSCSI), and objects (S3).

**Native S3 applications**

An increasing number of applications are able to leverage ONTAP support for object access using S3. Although well-suited for high-capacity archival workloads, the need for high performance in native S3 applications is growing rapidly and includes:
Customers can now use familiar manageability tools such as ONTAP System Manager to rapidly provision high-performance object storage for development and operations in ONTAP, taking advantage of the ONTAP storage efficiencies and security as they do so.

**FabricPool endpoints**

Beginning with ONTAP 9.8, FabricPool supports tiering to buckets in ONTAP, allowing for ONTAP-to-ONTAP tiering. This is an excellent option for customers who wish to repurpose existing FAS infrastructure as an object store endpoint.

FabricPool supports tiering to ONTAP in two ways:

- **Local cluster tiering.** Inactive data is tiered to a bucket located on the local cluster using cluster LIFs.
- **Remote cluster tiering.** Inactive data is tiered to a bucket located on a remote cluster in a manner similar to a traditional FabricPool cloud tier using IC LIFs on the FabricPool client and data LIFs on the ONTAP object store.

ONTAP S3 is appropriate if you want S3 capabilities on existing clusters without additional hardware and management. For deployments larger than 300TB, NetApp StorageGRID software continues to be the flagship NetApp solution for object storage. A FabricPool license is not required when using ONTAP or StorageGRID as the cloud tier.

**NetApp ONTAP for Confluent tiered storage**

Every data center needs to keep business-critical applications running and important data available and secure. The new NetApp AFF A900 system is powered by ONTAP Enterprise Edition software and a high-resilience design. Our new lightning-fast NVMe storage system eliminates disruptions to mission-critical operations, minimizes performance tuning, and safeguards your data from ransomware attacks.

From initial deployment to scaling your Confluent cluster, your environment demands rapid adaptation to changes that are nondisruptive to your business-critical applications. ONTAP enterprise data management, quality of service (QoS), and performance allow you to plan and adapt to your environment.

Using NetApp ONTAP and Confluent Tiered Storage together simplifies the management of Apache Kafka clusters by leveraging ONTAP as a scale-out storage target and enables independent scaling of compute and storage resources for Confluent.

An ONTAP S3 server is built on the mature scale-out storage capabilities of ONTAP. Scaling your ONTAP cluster can be performed seamlessly by extending your S3 buckets to use newly added nodes to the ONTAP cluster.

**Simple management with ONTAP System Manager**

ONTAP System Manager is a browser-based graphical interface that allows you to configure, manage, and monitor your ONTAP storage controller across globally distributed locations in a single pane of glass.
You can configure and manage ONTAP S3 with System Manager and the ONTAP CLI. When you enable S3 and create buckets using System Manager, ONTAP provides best-practice defaults for a simplified configuration. If you configure the S3 server and buckets from the CLI, you can still manage them with System Manager if desired or vice-versa.

When you create an S3 bucket using System Manager, ONTAP configures a default performance service level that is the highest available on your system. For example, on an AFF system, the default setting would be Extreme. Performance service levels are predefined adaptive QoS policy groups. Instead of one of the default service levels, you can specify a custom QoS policy group or no policy group.

Predefined adaptive QoS policy groups include the following:

- **Extreme.** Used for applications that require the lowest latency and highest performance.
- **Performance.** Used for applications with modest performance needs and latency.
- **Value.** Used for applications for which throughput and capacity are more important than latency.
- **Custom.** Specify a custom QoS policy or no QoS policy.

If you select **Use for tiering**, no performance service levels are selected, and the system tries to select low-cost media with optimal performance for the tiered data.

ONTAP tries to provision this bucket on local tiers that have the most appropriate disks, satisfying the chosen service level. However, if you need to specify which disks to include in the bucket, consider configuring S3 object storage from the CLI by specifying the local tiers (aggregate). If you configure the S3 server from the CLI, you can still manage it with System Manager if desired.

If you want the ability to specify which aggregates are used for buckets, you can only do so using the CLI.
Confluent

Confluent Platform is a full-scale data streaming platform that enables you to easily access, store, and manage data as continuous, real-time streams. Built by the original creators of Apache Kafka, Confluent expands the benefits of Kafka with enterprise-grade features while removing the burden of Kafka management or monitoring. Today, over 80% of the Fortune 100 are powered by data streaming technology, and most use Confluent.

Why Confluent?

By integrating historical and real-time data into a single, central source of truth, Confluent makes it easy to build an entirely new category of modern, event-driven applications, gain a universal data pipeline, and unlock powerful new use cases with full scalability, performance, and reliability.

What is Confluent used for?

Confluent Platform lets you focus on how to derive business value from your data rather than worrying about the underlying mechanics, such as how data is being transported or integrated between disparate systems. Specifically, Confluent Platform simplifies connecting data sources to Kafka, building streaming applications, as well as securing, monitoring, and managing your Kafka infrastructure. Today, Confluent Platform is used for a wide array of use cases across numerous industries, from financial services, omnichannel retail, and autonomous cars to fraud detection, microservices, and IoT.

The following figure shows the components of Confluent Platform.

Overview of Confluent event streaming technology

At the core of Confluent Platform is Kafka, the most popular open source distributed streaming platform. The key capabilities of Kafka include the following:

• Publish and subscribe to streams of records.
• Store streams of records in a fault tolerant way.
• Process streams of records.

Out of the box, Confluent Platform also includes Schema Registry, REST Proxy, a total of 100+ prebuilt Kafka connectors, and ksqlDB.

Overview of Confluent platform enterprise features

• **Confluent Control Center.** A UI-based system for managing and monitoring Kafka. It allows you to easily manage Kafka Connect and to create, edit, and manage connections to other systems.

• **Confluent for Kubernetes.** Confluent for Kubernetes is a Kubernetes operator. Kubernetes operators extend the orchestration capabilities of Kubernetes by providing the unique features and requirements for a specific platform application. For Confluent Platform, this includes greatly simplifying the deployment process of Kafka on Kubernetes and automating typical infrastructure lifecycle tasks.

• **Kafka Connect Connectors.** Connectors use the Kafka Connect API to connect Kafka to other systems such as databases, key-value stores, search indexes, and file systems. Confluent Hub has downloadable connectors for the most popular data sources and sinks, including fully tested and supported versions of these connectors with Confluent Platform. More details can be found here.

• **Self-balancing clusters.** Provides automated load balancing, failure detection and self-healing. It also provides support for adding or decommissioning brokers as needed, with no manual tuning.

• **Confluent cluster linking.** Directly connects clusters together and mirrors topics from one cluster to another over a link bridge. Cluster linking simplifies setup of multi-datacenter, multi-cluster, and hybrid cloud deployments.

• **Confluent auto data balancer.** Monitors your cluster for the number of brokers, the size of partitions, the number of partitions, and the number of leaders within the cluster. It allows you to shift data to create an even workload across your cluster, while throttling rebalance traffic to minimize the effect on production workloads while rebalancing.

• **Confluent replicator.** Makes it easier than ever to maintain multiple Kafka clusters in multiple data centers.

• **Tiered storage.** Provides options for storing large volumes of Kafka data using your favorite cloud provider, thereby reducing operational burden and cost. With tiered storage, you can keep data on cost-effective object storage and scale brokers only when you need more compute resources.

• **Confluent JMS client.** Confluent Platform includes a JMS-compatible client for Kafka. This Kafka client implements the JMS 1.1 standard API, using Kafka brokers as the backend. This is useful if you have legacy applications using JMS and you would like to replace the existing JMS message broker with Kafka.

• **Confluent MQTT proxy.** Provides a way to publish data directly to Kafka from MQTT devices and gateways without the need for a MQTT broker in the middle.

• **Confluent security plugins.** Confluent security plugins are used to add security capabilities to various Confluent Platform tools and products. Currently, there is a plugin available for the Confluent REST proxy that helps to authenticate the incoming requests and propagate the authenticated principal to requests to Kafka. This enables Confluent REST proxy clients to utilize the multitenant security features of the Kafka broker.

**Confluent Performance validation**

We have performed the verification with Confluent Platform for tiered storage on NetApp ONTAP. The NetApp and Confluent teams worked on this verification together and ran the test cases required for it.
**Confluent setup**

For the setup, we used three zookeepers, five brokers, and five testing servers with 256GB RAM and 16 CPUs. For NetApp storage, we used ONTAP with an AFF A900 HA pair. The storage and brokers were connected through 100GbE connections.

The following figure shows the network topology of configuration used for tiered storage verification.

The tools servers act as application clients that send or receive events to or from Confluent nodes.

**Confluent tiered storage configuration**

We used the following testing parameters:
confluent.tier.fetcher.num.threads=80
confluent.tier.archiver.num.threads=80
confluent.tier.enable=true
confluent.tier.feature=true
confluent.tier.backend=S3
confluent.tier.s3.bucket=kafkabucket1-1
confluent.tier.s3.region=us-east-1
confluent.tier.s3.cred.file.path=/data/kafka/.ssh/credentials
confluent.tier.s3.aws.endpoint.override=http://wle-mendocino-07-08/
confluent.tier.s3.force.path.style.access=true
debug=true
jmx.port=7203
num.partitions=80
num.records=20000000
segment.bytes=536870912
max.partition.fetch.bytes=1048576000
length.key.value=2048
trogdor.agent.nodes=node0,node1,node2,node3,node4
trogdor.coordinator.hostname.port=192.168.150.155:8889
num.producers=20
num.head.consumers=20
num.tail.consumers=1
test.binary.task.max.heap.size=32G
test.binary.task.timeout.sec=3600
producer.timeout.sec=3600
consumer.timeout.sec=3600

For verification, we used ONTAP with the HTTP protocol, but HTTPS also worked. The access key and secret key are stored in the file name provided in the confluent.tier.s3.cred.file.path parameter.

**NetApp storage controller – ONTAP**

We configured a single HA pair configuration in ONTAP for verification.
Verification results

We completed the following five test cases for the verification. The first two were functionality tests and the remaining three were performance tests.

Object store correctness test

This test performs basic operations such as get, put, and delete on the object store used for the tiered storage using API calls.
**Tiering functionality correctness test**

This test checks the end-to-end functionality of the object storage. It creates a topic, produces an event stream to the newly created topic, waits for the brokers to archive the segments to the object storage, consumes the event stream, and validates the consumed stream matches with the produced stream. We have performed this test with and without an object-store fault injection. We simulated node failure by stopping the service manager service in one of the nodes in ONTAP and validating that the end-to-end functionality works with object storage.

**Tier fetch benchmark**

This test validated the read performance of the tiered object storage and checked the range fetch read requests under heavy load from segments generated by the benchmark. In this benchmark, Confluent developed custom clients to serve the tier fetch requests.

**Produce-consume workload generator**

This test indirectly generates write workload on the object store through the archival of segments. The read workload (segments read) was generated from object storage when consumer groups fetched the segments. This workload was generated by a TOCC script. This test checked the performance of read and write on the object storage in parallel threads. We tested with and without object store fault injection as we did for the tiering functionality correctness test.

**Retention workload generator**

This test checked the deletion performance of an object storage under a heavy topic-retention workload. The retention workload was generated using a TOCC script that produces many messages in parallel to a test topic. The test topic was configuring with an aggressive size-based and time-based retention setting that caused the event stream to be continuously purged from the object store. The segments were then archived. This led to many deletions in the object storage by the broker and collection of the performance of the object-store delete operations.

For verification details, see the Confluent website.

**Performance tests with produce-consume workload generator**

We performed tiered storage testing with either five or eight broker nodes during a produce-consume workload with the one AFF A900 HA pair NetApp storage controller. According to our tests, the time to completion and the performance results scaled with the number of broker nodes until AFF A900 resource utilization reached one hundred percent. The ONTAP storage controller setup required a minimum of one HA pair.

The performance for the S3 retrieve operation increased linearly based on number of Confluent broker nodes. ONTAP storage controller supports up to 12 HA pair in a single deployment.

The following graph shows combined S3 tiering traffic with five or eight broker nodes. We maximized the AFF A900 single HA pair performance.
The following graph shows the Kafka throughput at approximately 31.74GBps.

We also observed similar throughput in the ONTAP storage controller perfstat report.

```
object_store_server:wle-mendocino-07-08:get_data:34080805907b/ s
object_store_server:wle-mendocino-07-08:put_data:484236974b/ s
```

Performance best practice guidelines

This page describes the best practices for improving performance in this solution.

- For ONTAP, when possible, use a GET size >=1MB.
- Increasing `num.network.threads` and `num.io.threads` in `server.properties` on broker nodes enables you to push increased tiering activity to S3 tier. These results are with `num.network.threads` and `num.io.threads` set to 32.
• S3 buckets should target eight constituents per member aggregate.
• Ethernet links driving S3 traffic should use an MTU of 9k when possible on both storage and client.

Conclusion

This verification test reached 31.74GBps of tiering throughput on Confluent with NetApp ONTAP Storage Controller.

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

• What is Confluent?
  https://www.confluent.io/apache-kafka-vs-confluent/

• S3-sink parameter details
  https://docs.confluent.io/kafka-connect-s3-sink/current/configuration_options.html#s3-configuration-options

• Apache Kafka
  https://en.wikipedia.org/wiki/Apache_Kafka

• S3 in ONTAP best practices

• S3 Object storage Management
  https://docs.netapp.com/us-en/ontap/s3-config/s3-support-concept.html

• NetApp Product Documentation
  https://www.netapp.com/support-and-training/documentation/

NetApp Storage Solutions for Apache Spark

TR-4570: NetApp Storage Solutions for Apache Spark: Architecture, Use Cases, and Performance Results

Rick Huang, Karthikeyan Nagalingam, NetApp

This document focuses on the Apache Spark architecture, customer use cases, and the NetApp storage portfolio related to big data analytics and artificial intelligence (AI). It also presents various testing results using industry-standard AI, machine learning (ML), and deep learning (DL) tools against a typical Hadoop system so that you can choose the appropriate Spark solution. To begin, you need a Spark architecture, appropriate components, and two deployment modes (cluster and client).
This document also provides customer use cases to address configuration issues, and it discusses an overview of the NetApp storage portfolio relevant to big data analytics and AI, ML, and DL with Spark. We then finish with testing results derived from Spark-specific use cases and the NetApp Spark solution portfolio.

**Customer challenges**

This section focuses on customer challenges with big data analytics and AI/ML/DL in data growth industries such as retail, digital marketing, banking, discrete manufacturing, process manufacturing, government, and professional services.

**Unpredictable performance**

Traditional Hadoop deployments typically use commodity hardware. To improve performance, you must tune the network, operating system, Hadoop cluster, ecosystem components such as Spark, and hardware. Even if you tune each layer, it can be difficult to achieve desired performance levels because Hadoop is running on commodity hardware that was not designed for high performance in your environment.

**Media and node failures**

Even under normal conditions, commodity hardware is prone to failure. If one disk on a data node fails, the Hadoop master by default considers that node to be unhealthy. It then copies specific data from that node over the network from replicas to a healthy node. This process slows down the network packets for any Hadoop jobs. The cluster must then copy the data back again and remove the over-replicated data when the unhealthy node returns to a healthy state.

**Hadoop vendor lock-in**

Hadoop distributors have their own Hadoop distribution with their own versioning, which locks in the customer to those distributions. However, many customers require support for in-memory analytics that does not tie the customer to specific Hadoop distributions. They need the freedom to change distributions and still bring their analytics with them.

**Lack of support for more than one language**

Customers often require support for multiple languages in addition to MapReduce Java programs to run their jobs. Options such as SQL and scripts provide more flexibility for getting answers, more options for organizing and retrieving data, and faster ways of moving data into an analytics framework.

**Difficulty of use**

For some time, people have complained that Hadoop is difficult to use. Even though Hadoop has become simpler and more powerful with each new version, this critique has persisted. Hadoop requires that you understand Java and MapReduce programming patterns, a challenge for database administrators and people with traditional scripting skill sets.

**Complicated frameworks and tools**

Enterprises AI teams face multiple challenges. Even with expert data science knowledge, tools and frameworks for different deployment ecosystems and applications might not translate simply from one to another. A data science platform should integrate seamlessly with corresponding big data platforms built on Spark with ease of data movement, reusable models, code out of the box, and tools that support best practices for prototyping, validating, versioning, sharing, reusing, and quickly deploying models to production.
Why choose NetApp?

NetApp can improve your Spark experience in the following ways:

- NetApp NFS direct access (shown in the figure below) allows customers to run big-data-analytics jobs on their existing or new NFSv3 or NFSv4 data without moving or copying the data. It prevents multiple copies of data and eliminates the need to sync the data with a source.
- More efficient storage and less server replication. For example, the NetApp E-Series Hadoop solution requires two rather than three replicas of the data, and the FAS Hadoop solution requires a data source but no replication or copies of data. NetApp storage solutions also produce less server-to-server traffic.
- Better Hadoop job and cluster behavior during drive and node failure.
- Better data-ingest performance.

For example, in the financial and healthcare sector, the movement of data from one place to another must meet legal obligations, which is not an easy task. In this scenario, NetApp NFS direct access analyzes the financial and healthcare data from its original location. Another key benefit is that using NetApp NFS direct access simplifies protecting Hadoop data by using native Hadoop commands and enabling data protection workflows with the rich data management portfolio from NetApp.

NetApp NFS direct access provides two kinds of deployment options for Hadoop/Spark clusters:

- By default, Hadoop or Spark clusters use the Hadoop Distributed File System (HDFS) for data storage and the default file system. NetApp NFS direct access can replace the default HDFS with NFS storage as the default file system, enabling direct analytics on NFS data.
- In another deployment option, NetApp NFS direct access supports configuring NFS as additional storage along with HDFS in a single Hadoop or Spark cluster. In this case, the customer can share data through NFS exports and access it from the same cluster along with HDFS data.

The key benefits of using NetApp NFS direct access include the following:

- Analyzing the data from its current location, which prevents the time- and performance-consuming task of
moving analytics data to a Hadoop infrastructure such as HDFS.

• Reducing the number of replicas from three to one.
• Enabling users to decouple compute and storage to scale them independently.
• Providing enterprise data protection by leveraging the rich data management capabilities of ONTAP.
• Certification with the Hortonworks data platform.
• Enabling hybrid data analytics deployments.
• Reducing backup time by leveraging dynamic multithread capability.

See TR-4657: NetApp hybrid cloud data solutions - Spark and Hadoop based on customer use cases for backing up Hadoop data, backup and disaster recovery from the cloud to on-premises, enabling DevTest on existing Hadoop data, data protection and multicloud connectivity, and accelerating analytics workloads.

The following sections describe storage capabilities that are important for Spark customers.

Storage tiering

With Hadoop storage tiering, you can store files with different storage types in accordance with a storage policy. Storage types include hot, cold, warm, all_ssd, one_ssd, and lazy_persist.

-------- HEAD

We performed validation of Hadoop storage tiering on a NetApp AFF storage controller and an E-Series storage controller with SSD and SAS drives with different storage policies. The Spark cluster with AFF-A800 has four compute worker nodes, whereas the cluster with E-Series has eight. This is mainly to compare the performance of solid-state drives (SSDs) versus hard-drive disks (HDDs).
We performed validation of Hadoop storage tiering on a NetApp AFF storage controller and an E-Series storage controller with SSD and SAS drives with different storage policies. The Spark cluster with AFF-A800 has four compute worker nodes, whereas the cluster with E-Series has eight. We did this primarily to compare the performance of solid-state drives to hard-drive disks.

The following figure shows the performance of NetApp solutions for a Hadoop SSD.

- The baseline NL-SAS configuration used eight compute nodes and 96 NL-SAS drives. This configuration generated 1TB of data in 4 minutes and 38 seconds. See TR-3969 NetApp E-Series Solution for Hadoop for details on the cluster and storage configuration.

- Using TeraGen, the SSD configuration generated 1TB of data 15.66x faster than the NL-SAS configuration. Moreover, the SSD configuration used half the number of compute nodes and half the number of disk drives (24 SSD drives in total). Based on the job completion time, it was almost twice as fast as the NL-SAS configuration.

- Using TeraSort, the SSD configuration sorted 1TB of data 1138.36 times more quickly than the NL-SAS configuration. Moreover, the SSD configuration used half the number of compute nodes and half the number of disk drives (24 SSD drives in total). Therefore, per drive, it was approximately three times faster than the NL-SAS configuration.

- The takeaway is transitioning from spinning disks to all-flash improves performance. The number of compute nodes was not the bottleneck. With NetApp’s all-flash storage, runtime performance scales well.

- With NFS, the data was functionally equivalent to being pooled all together, which can reduce the number of compute nodes depending on your workload. The Apache Spark cluster users do not have to manually rebalance data when changing number of compute nodes.

- In summary, transitioning from spinning disks to all-flash improves performance. The number of compute nodes was not the bottleneck. With NetApp all-flash storage, runtime performance scales well.
• With NFS, data was functionally equivalent to being pooled all together, which can reduce the number of compute nodes depending on your workload. Apache Spark cluster users do not need to manually rebalance data when changing the number of compute nodes.

Performance scaling - Scale out

When you need more computation power from a Hadoop cluster in an AFF solution, you can add data nodes with an appropriate number of storage controllers. NetApp recommends starting with four data nodes per storage controller array and increasing the number to eight data nodes per storage controller, depending on workload characteristics.

AFF and FAS are perfect for in-place analytics. Based on computation requirements, you can add node managers, and non-disruptive operations allow you to add a storage controller on demand without downtime. We offer rich features with AFF and FAS, such as NVME media support, guaranteed efficiency, data reduction, QOS, predictive analytics, cloud tiering, replication, cloud deployment, and security. To help customers meet their requirements, NetApp offers features such as file system analytics, quotas, and on-box load balancing with no additional license costs. NetApp has better performance in the number of concurrent jobs, lower latency, simpler operations, and higher gigabytes per second throughput than our competitors. Furthermore, NetApp Cloud Volumes ONTAP runs on all three major cloud providers.

Performance scaling - Scale up

Scale-up features allow you to add disk drives to AFF, FAS, and E-Series systems when you need additional storage capacity. With Cloud Volumes ONTAP, scaling storage to the PB level is a combination of two factors: tiering infrequently used data to object storage from block storage and stacking Cloud Volumes ONTAP licenses without additional compute.

Multiple protocols

NetApp systems support most protocols for Hadoop deployments, including SAS, iSCSI, FCP, InfiniBand, and NFS.

Operational and supported solutions

The Hadoop solutions described in this document are supported by NetApp. These solutions are also certified with major Hadoop distributors. For information, see the MapR site, the Hortonworks site, and the Cloudera certification and partner sites.

Target audience

The world of analytics and data science touches multiple disciplines in IT and business:

• The data scientist needs the flexibility to use their tools and libraries of choice.
• The data engineer needs to know how the data flows and where it resides.
• A DevOps engineer needs the tools to integrate new AI and ML applications into their CI and CD pipelines.
• Cloud administrators and architects must be able to set up and manage hybrid cloud resources.
• Business users want to have access to analytics, AI, ML, and DL applications.

In this technical report, we describe how NetApp AFF, E-Series, StorageGRID, NFS direct access, Apache Spark, Horovod, and Keras help each of these roles bring value to business.
Solution technology

Apache Spark is a popular programming framework for writing Hadoop applications that works directly with the Hadoop Distributed File System (HDFS). Spark is production ready, supports processing of streaming data, and is faster than MapReduce. Spark has configurable in-memory data caching for efficient iteration, and the Spark shell is interactive for learning and exploring data. With Spark, you can create applications in Python, Scala, or Java. Spark applications consist of one or more jobs that have one or more tasks.

Every Spark application has a Spark driver. In YARN-Client mode, the driver runs on the client locally. In YARN-Cluster mode, the driver runs in the cluster on the application master. In the cluster mode, the application continues to run even if the client disconnects.

There are three cluster managers:

- **Standalone.** This manager is a part of Spark, which makes it easy to set up a cluster.
- **Apache Mesos.** This is a general cluster manager that also runs MapReduce and other applications.
- **Hadoop YARN.** This is a resource manager in Hadoop 3.
The resilient distributed dataset (RDD) is the primary component of Spark. RDD recreates the lost and missing data from data stored in memory in the cluster and stores the initial data that comes from a file or is created programmatically. RDDs are created from files, data in memory, or another RDD. Spark programming performs two operations: transformation and actions. Transformation creates a new RDD based on an existing one. Actions return a value from an RDD.

Transformations and actions also apply to Spark Datasets and DataFrames. A dataset is a distributed collection of data that provides the benefits of RDDs (strong typing, use of lambda functions) with the benefits of Spark SQL’s optimized execution engine. A Dataset can be constructed from JVM objects and then manipulated using functional transformations (map, flatMap, filter, and so on.). A DataFrame is a dataset organized into named columns. It is conceptually equivalent to a table in a relational database or a data frame in R/Python. DataFrames can be constructed from a wide array of sources such as structured data files, tables in Hive/HBase, external databases on-premises or in the cloud, or existing RDDs.

Spark applications include one or more Spark jobs. Jobs run tasks in executors, and executors run in YARN containers. Each executor runs in a single container, and executors exist throughout the life of an application. An executor is fixed after the application starts, and YARN does not resize the already allocated container. An executor can run tasks concurrently on in-memory data.

**NetApp Spark solutions overview**

NetApp has three storage portfolios: FAS/AFF, E-Series, and Cloud Volumes ONTAP. We have validated AFF and the E-Series with ONTAP storage system for Hadoop solutions with Apache Spark.

The data fabric powered by NetApp integrates data management services and applications (building blocks) for data access, control, protection, and security, as shown in the figure below.

The building blocks in the figure above include:

- **NetApp NFS direct access.** Provides the latest Hadoop and Spark clusters with direct access to NetApp NFS volumes without additional software or driver requirements.

- **NetApp Cloud Volumes ONTAP and Cloud Volume Services.** Software-defined connected storage based on ONTAP running in Amazon Web Services (AWS) or Azure NetApp Files (ANF) in Microsoft Azure.
cloud services.

- **NetApp SnapMirror technology.** Provides data protection capabilities between on-premises and ONTAP Cloud or NPS instances.

- **Cloud service providers.** These providers include AWS, Microsoft Azure, Google Cloud, and IBM Cloud.

- **PaaS.** Cloud-based analytics services such as Amazon Elastic MapReduce (EMR) and Databricks in AWS as well as Microsoft Azure HDInsight and Azure Databricks.

The following figure depicts the Spark solution with NetApp storage.

![Spark Solution with NetApp Storage Diagram](image)

The ONTAP Spark solution uses the NetApp NFS direct access protocol for in-place analytics and AI, ML, and DL workflows using access to existing production data. Production data available to Hadoop nodes is exported to perform in-place analytical and AI, ML, and DL jobs. You can access data to process in Hadoop nodes either with NetApp NFS direct access or without it. In Spark with the standalone or yarn cluster manager, you can configure an NFS volume by using `file:///<target_volume>`. We validated three use cases with different datasets. The details of these validations are presented in the section “Testing Results.” (xref)
We identified the unique features of the E-Series Spark solution, the AFF/FAS ONTAP Spark solution, and the StorageGRID Spark solution, and performed detailed validation and testing. Based upon our observations, NetApp recommends the E-Series solution for greenfield installations and new scalable deployments and the AFF/FAS solution for in-place analytics, AI, ML, and DL workloads using existing NFS data, and StorageGRID for AI, ML, and DL and modern data analytics when object storage is required.

A data lake is a storage repository for large datasets in native form that can be used for analytics, AI, ML, and DL jobs. We built a data lake repository for the E-Series, AFF/FAS, and StorageGRID SG6060 Spark solutions. The E-Series system provides HDFS access to the Hadoop Spark cluster, whereas existing production data is accessed through the NFS direct access protocol to the Hadoop cluster. For datasets that reside in object storage, NetApp StorageGRID provides S3 and S3a secure access.

Use case summary

This page describes the different areas in which this solution can be used.

Streaming data

Apache Spark can process streaming data, which is used for streaming extract, transform, and load (ETL) processes; data enrichment; triggering event detection; and complex session analysis:

- **Streaming ETL.** Data is continually cleaned and aggregated before it is pushed into datastores. Netflix uses Kafka and Spark streaming to build a real-time online movie recommendation and data monitoring solution that can process billions of events per day from different data sources. Traditional ETL for batch processing is treated differently, however. This data is read first, and then it is converted into a database format before being written to the database.

- **Data enrichment.** Spark streaming enriches the live data with static data to enable more real-time data analysis. For example, online advertisers can deliver personalized, targeted ads directed by information about customer behavior.

- **Trigger event detection.** Spark streaming allows you to detect and respond quickly to unusual behavior that could indicate potentially serious problems. For example, financial institutions use triggers to detect and stop fraud transactions, and hospitals use triggers to detect dangerous health changes detected in a patient’s vital signs.

- **Complex session analysis.** Spark streaming collects events such as user activity after logging in to a...
website or application, which are then grouped and analyzed. For example, Netflix uses this functionality to provide real-time movie recommendations.

For more streaming data configuration, Confluent Kafka verification, and performance tests, see TR-4912: Best practice guidelines for Confluent Kafka tiered storage with NetApp.

**Machine learning**

The Spark integrated framework helps you run repeated queries on datasets using the machine learning library (MLlib). MLlib is used in areas such as clustering, classification, and dimensionality reduction for some common big data functions such as predictive intelligence, customer segmentation for marketing purposes, and sentiment analysis. MLlib is used in network security to conduct real-time inspections of data packets for indications of malicious activity. It helps security providers learn about new threats and stay ahead of hackers while protecting their clients in real time.

**Deep learning**

TensorFlow is a popular deep learning framework used across the industry. TensorFlow supports the distributed training on a CPU or GPU cluster. This distributed training allows users to run it on a large amount of data with lot of deep layers.

Until fair recently, if we wanted to use TensorFlow with Apache Spark, we needed to perform all necessary ETL for TensorFlow in PySpark and then write data to intermediate storage. That data would then be loaded onto the TensorFlow cluster for the actual training process. This workflow required the user to maintain two different clusters, one for ETL and one for distributed training of TensorFlow. Running and maintaining multiple clusters was typically tedious and time consuming.

DataFrames and RDD in earlier Spark versions were not well-suited for deep learning because random access was limited. In Spark 3.0 with project hydrogen, native support for the deep learning frameworks is added. This approach allows non-MapReduce-based scheduling on the Spark cluster.

**Interactive analysis**

Apache Spark is fast enough to perform exploratory queries without sampling with development languages other than Spark, including SQL, R, and Python. Spark uses visualization tools to process complex data and visualize it interactively. Spark with structured streaming performs interactive queries against live data in web analytics that enable you to run interactive queries against a web visitor’s current session.

**Recommender system**

Over the years, recommender systems have brought tremendous changes to our lives, as businesses and consumers have responded to dramatic changes in online shopping, online entertainment, and many other industries. Indeed, these systems are among the most evident success stories of AI in production. In many practical use cases, recommender systems are combined with conversational AI or chatbots interfaced with an NLP backend to obtain relevant information and produce useful inferences.

Today, many retailers are adopting newer business models like buying online and picking up in store, curbside pickup, self-checkout, scan-and-go, and more. These models have become prominent during the COVID-19 pandemic by making shopping safer and more convenient for consumers. AI is crucial for these growing digital trends, which are influenced by consumer behavior and vice versa. To meet the growing demands of consumers, to augment the customer experience, to improve operational efficiency, and to grow revenue, NetApp helps its enterprise customers and businesses use machine-learning and deep-learning algorithms to design faster and more accurate recommender systems.

There are several popular techniques used for providing recommendations, including collaborative filtering,
content-based systems, the deep learning recommender model (DLRM), and hybrid techniques. Customers previously utilized PySpark to implement collaborative filtering for creating recommendation systems. Spark MLlib implements alternating least squares (ALS) for collaborative filtering, a very popular algorithm among enterprises before the rise of DLRM.

**Natural language processing**

Conversational AI, made possible by natural language processing (NLP), is the branch of AI helping computers communicate with humans. NLP is prevalent in every industry vertical and many use cases, from smart assistants and chatbots to Google search and predictive text. According to a Gartner prediction, by 2022, 70% of people will be interacting with conversational AI platforms on a daily basis. For a high-quality conversation between a human and a machine, responses must be rapid, intelligent, and natural sounding.

Customers need a large amount of data to process and train their NLP and automatic speech recognition (ASR) models. They also need to move data across the edge, core, and cloud, and they need the power to perform inference in milliseconds to establish natural communication with humans. NetApp AI and Apache Spark is an ideal combination for compute, storage, data processing, model training, fine-tuning, and deployment.

Sentiment analysis is a field of study within NLP in which positive, negative, or neutral sentiments are extracted from text. Sentiment analysis has a variety of use cases, from determining support center employee performance in conversations with callers to providing appropriate automated chatbot responses. It has also been used to predict a firm’s stock price based on the interactions between firm representatives and the audience at quarterly earnings calls. Furthermore, sentiment analysis can be used to determine a customer’s view on the products, services, or support provided by the brand.

We used the Spark NLP library from John Snow Labs to load pretrained pipelines and Bidirectional Encoder Representations from Transformers (BERT) models including financial news sentiment and FinBERT, performing tokenization, named entity recognition, model training, fitting and sentiment analysis at scale. Spark NLP is the only open-source NLP library in production that offers state-of-the-art transformers such as BERT, ALBERT, ELECTRA, XLNet, DistilBERT, RoBERTa, DeBERTa, XLM- RoBERTa, Longformer, ELMO, Universal Sentence Encoder, Google T5, MarianMT, and GPT2. The library works not only in Python and R, but also in the JVM ecosystem (Java, Scala, and Kotlin) at scale by extending Apache Spark natively.

**Major AI, ML, and DL use cases and architectures**

Major AI, ML, and DL use cases and methodology can be divided into the following sections:

**Spark NLP pipelines and TensorFlow distributed inferencing**

The following list contains the most popular open-source NLP libraries that have been adopted by the data science community under different levels of development:

- **Natural Language Toolkit (NLTK)**. The complete toolkit for all NLP techniques. It has been maintained since the early 2000s.
- **TextBlob**. An easy-to-use NLP tools Python API built on top of NLTK and Pattern.
- **Stanford Core NLP**. NLP services and packages in Java developed by the Stanford NLP Group.
- **Gensim**. Topic Modelling for Humans started off as a collection of Python scripts for the Czech Digital Mathematics Library project.
- **SpaCy**. End-to-end industrial NLP workflows with Python and Cython with GPU acceleration for transformers.
• Fasttext. A free, lightweight, open-source NLP library for the learning-of-word embeddings and sentence classification created by Facebook’s AI Research (FAIR) lab.

Spark NLP is a single, unified solution for all NLP tasks and requirements that enables scalable, high-performance, and high-accuracy NLP-powered software for real production use cases. It leverages transfer learning and implements the latest state-of-the-art algorithms and models in research and across industries. Due to the lack of full support by Spark for the above libraries, Spark NLP was built on top of Spark ML to take advantage of Spark’s general-purpose in-memory distributed data processing engine as an enterprise-grade NLP library for mission-critical production workflows. Its annotators utilize rule-based algorithms, machine learning, and TensorFlow to power deep learning implementations. This covers common NLP tasks including but not limited to tokenization, lemmatization, stemming, part-of-speech tagging, named-entity recognition, spell checking, and sentiment analysis.

Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based machine learning technique for NLP. It popularized the concept of pretraining and fine tuning. The transformer architecture in BERT originated from machine translation, which models long-term dependencies better than Recurrent Neural Network (RNN)-based language models. It also introduced the Masked Language Modelling (MLM) task, where a random 15% of all tokens are masked and the model predicts them, enabling true bidirectionality.

Financial sentiment analysis is challenging due to the specialized language and lack of labeled data in that domain. FinBERT, a language model based on pretrained BERT, was domain adapted onReuters TRC2, a financial corpus, and fine-tuned with labeled data (Financial PhraseBank) for financial sentiment classification. Researchers extracted 4,500 sentences from news articles with financial terms. Then 16 experts and masters students with finance backgrounds labeled the sentences as positive, neutral, and negative. We built an end-to-end Spark workflow to analyze sentiment for Top-10 NASDAQ company earnings call transcripts from 2016 to 2020 using FinBERT and two other pre-trained pipelines (Sentiment Analysis for Financial News, Explain Document DL) from Spark NLP.

The underlying deep learning engine for Spark NLP is TensorFlow, an end-to-end, open-source platform for machine learning that enables easy model building, robust ML production anywhere, and powerful experimentation for research. Therefore, when executing our pipelines in Spark yarn cluster mode, we were essentially running distributed TensorFlow with data and model parallelization across one master and multiple worker nodes, as well as network-attached storage mounted on the cluster.

Horovod distributed training

The core Hadoop validation for MapReduce-related performance is performed with TeraGen, TeraSort, TeraValidate, and DFSIO (read and write). The TeraGen and TeraSort validation results are presented in TR-3969: NetApp Solutions for Hadoop for E-Series and in the section “Storage Tiering” (xref) for AFF.

Based upon customer requests, we consider distributed training with Spark to be one of the most important of the various use cases. In this document, we used the Horovod on Spark to validate Spark performance with NetApp on-premises, cloud-native, and hybrid cloud solutions using NetApp All Flash FAS (AFF) storage controllers, Azure NetApp Files, and StorageGRID.

The Horovod on Spark package provides a convenient wrapper around Horovod that makes running distributed training workloads in Spark clusters simple, enabling a tight model design loop in which data processing, model training, and model evaluation are all done in Spark where training and inferencing data resides.

There are two APIs for running Horovod on Spark: a high-level Estimator API and a lower-level Run API. Although both use the same underlying mechanism to launch Horovod on Spark executors, the Estimator API abstracts the data processing, model training loop, model checkpointing, metrics collection, and distributed training. We used Horovod Spark Estimators, TensorFlow, and Keras for an end-to-end data preparation and distributed training workflow based on the Kaggle Rossmann Store Sales competition.
The script `keras_spark_horovod_rossmann_estimator.py` can be found in the section "Python scripts for each major use case." It contains three parts:

- The first part performs various data preprocessing steps over an initial set of CSV files provided by Kaggle and gathered by the community. The input data is separated into a training set with a Validation subset, and a testing dataset.
- The second part defines a Keras Deep Neural Network (DNN) model with logarithmic sigmoid activation function and an Adam optimizer, and it performs distributed training of the model using Horovod on Spark.
- The third part performs prediction on the testing dataset using the best model that minimizes the validation set overall mean absolute error. It then creates an output CSV file.

See the section “Machine Learning” for various runtime comparison results.

Multi-worker deep learning using Keras for CTR prediction

With the recent advances in ML platforms and applications, a lot of attention is now on learning at scale. The click-through rate (CTR) is defined as the average number of click-throughs per hundred online ad impressions (expressed as a percentage). It is widely adopted as a key metric in various industry verticals and use cases, including digital marketing, retail, e-commerce, and service providers. See our TR-4904: Distributed training in Azure - Click-Through Rate Prediction for more detail on the applications of CTR and an end-to-end Cloud AI workflow implementation with Kubernetes, distributed data ETL, and model training using Dask and CUDA ML.

In this technical report we used a variation of the Criteo Terabyte Click Logs dataset (see TR-4904) for multi-worker distributed deep learning using Keras to build a Spark workflow with Deep and Cross Network (DCN) models, comparing its performance in terms of log loss error function with a baseline Spark ML Logistic Regression model. DCN efficiently captures effective feature interactions of bounded degrees, learns highly nonlinear interactions, requires no manual feature engineering or exhaustive searching, and has low computational cost.

Data for web-scale recommender systems is mostly discrete and categorical, leading to a large and sparse feature space that is challenging for feature exploration. This has limited most large-scale systems to linear models such as logistic regression. However, identifying frequently predictive features and at the same time exploring unseen or rare cross features is the key to making good predictions. Linear models are simple, interpretable, and easy to scale, but they are limited in their expressive power.

Cross features, on the other hand, have been shown to be significant in improving the models’ expressiveness. Unfortunately, it often requires manual feature engineering or exhaustive search to identify such features. Generalizing to unseen feature interactions is often difficult. Using a cross neural network like DCN avoids task-specific feature engineering by explicitly applying feature crossing in an automatic fashion. The cross network consists of multiple layers, where the highest degree of interactions is provably determined by layer depth. Each layer produces higher-order interactions based on existing ones and keeps the interactions from previous layers.

A deep neural network (DNN) has the promise to capture very complex interactions across features. However, compared to DCN, it requires nearly an order of magnitude more parameters, is unable to form cross features explicitly, and may fail to efficiently learn some types of feature interactions. The cross network is memory efficient and easy to implement. Jointly training the cross and DNN components together efficiently captures predictive feature interactions and delivers state-of-the-art performance on the Criteo CTR dataset.

A DCN model starts with an embedding and stacking layer, followed by a cross network and a deep network in parallel. These in turn are followed by a final combination layer which combines the outputs from the two networks. Your input data can be a vector with sparse and dense features. In Spark, both ml and mllib libraries contain the type SparseVector. It is therefore important for users to distinguish between the two and be mindful when calling their respective functions and methods. In web-scale recommender systems such as CTR
prediction, the inputs are mostly categorical features, for example 'country=usa'. Such features are often encoded as one-hot vectors, for example, '[0,1,0, ...]'. One-hot-encoding (OHE) with `SparseVector` is useful when dealing with real-world datasets with ever-changing and growing vocabularies. We modified examples in `DeepCTR` to process large vocabularies, creating embedding vectors in the embedding and stacking layer of our DCN.

The Criteo Display Ads dataset predicts the ads click-through rate. It has 13 integer features and 26 categorical features in which each category has a high cardinality. For this dataset, an improvement of 0.001 in logloss is practically significant due to the large input size. A small improvement in prediction accuracy for a large user base can potentially lead to a large increase in a company's revenue. The dataset contains 11GB of user logs from a period of 7 days, which equates to around 41 million records. We used Spark `dataFrame.randomSplit()` function to randomly split the data for training (80%), cross-validation (10%), and the remaining 10% for testing.

DCN was implemented on TensorFlow with Keras. There are four main components in implementing the model training process with DCN:

- **Data processing and embedding.** Real-valued features are normalized by applying a log transform. For categorical features, we embed the features in dense vectors of dimension $6 \times (\text{category cardinality})^{1/4}$. Concatenating all embeddings results in a vector of dimension 1026.

- **Optimization.** We applied mini-batch stochastic optimization with the Adam optimizer. The batch size was set to 512. Batch normalization was applied to the deep network and the gradient clip norm was set at 100.

- **Regularization.** We used early stopping, as L2 regularization or dropout was not found to be effective.

- **Hyperparameters.** We report results based on a grid search over the number of hidden layers, the hidden layer size, the initial learning rate, and the number of cross layers. The number of hidden layers ranged from 2 to 5, with hidden layer sizes ranging from 32 to 1024. For DCN, the number of cross layers was from 1 to 6. The initial learning rate was tuned from 0.0001 to 0.001 with increments of 0.0001. All experiments applied early stopping at training step 150,000, beyond which overfitting started to occur.

In addition to DCN, we also tested other popular deep-learning models for CTR prediction, including `DeepFM`, `xDeepFM`, `AutoInt`, and `DCN v2`.

**Architectures used for validation**

For this validation, we used four worker nodes and one master nodes with an AFF-A800 HA pair. All cluster members were connected through 10GbE network switches.

For this NetApp Spark solution validation, we used three different storage controllers: the E5760, the E5724, and the AFF-A800. The E-Series storage controllers were connected to five data nodes with 12Gbps SAS connections. The AFF HA-pair storage controller provides exported NFS volumes through 10GbE connections to Hadoop worker nodes. The Hadoop cluster members were connected through 10GbE connections in the E-Series, AFF, and StorageGRID Hadoop solutions.
Testing results

We used the TeraSort and TeraValidate scripts in the TeraGen benchmarking tool to
measure the Spark performance validation with E5760, E5724, and AFF-A800 configurations. In addition, three major use cases were tested: Spark NLP pipelines and TensorFlow distributed training, Horovod distributed training, and multi-worker deep learning using Keras for CTR Prediction with DeepFM.

For both E-Series and StorageGRID validation, we used Hadoop replication factor 2. For AFF validation, we only used one source of data.

The following table lists the hardware configuration for the Spark performance validation.

<table>
<thead>
<tr>
<th>Type</th>
<th>Hadoop worker nodes</th>
<th>Drive type</th>
<th>Drives per node</th>
<th>Storage controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG6060</td>
<td>4</td>
<td>SAS</td>
<td>12</td>
<td>Single high-availability (HA) pair</td>
</tr>
<tr>
<td>E5760</td>
<td>4</td>
<td>SAS</td>
<td>60</td>
<td>Single HA pair</td>
</tr>
<tr>
<td>E5724</td>
<td>4</td>
<td>SAS</td>
<td>24</td>
<td>Single HA pair</td>
</tr>
<tr>
<td>AFF800</td>
<td>4</td>
<td>SSD</td>
<td>6</td>
<td>Single HA pair</td>
</tr>
</tbody>
</table>

The following table lists software requirements.

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHEL</td>
<td>7.9</td>
</tr>
<tr>
<td>OpenJDK Runtime Env</td>
<td>1.8.0</td>
</tr>
<tr>
<td>OpenJDK 64-Bit Server VM</td>
<td>25.302</td>
</tr>
<tr>
<td>Git</td>
<td>2.24.1</td>
</tr>
<tr>
<td>GCC/G++</td>
<td>11.2.1</td>
</tr>
<tr>
<td>Spark</td>
<td>3.2.1</td>
</tr>
<tr>
<td>PySpark</td>
<td>3.1.2</td>
</tr>
<tr>
<td>SparkNLP</td>
<td>3.4.2</td>
</tr>
<tr>
<td>TensorFlow</td>
<td>2.9.0</td>
</tr>
<tr>
<td>Keras</td>
<td>2.9.0</td>
</tr>
<tr>
<td>Horovod</td>
<td>0.24.3</td>
</tr>
</tbody>
</table>

**Financial sentiment analysis**

We published TR-4910: Sentiment Analysis from Customer Communications with NetApp AI, in which an end-to-end conversational AI pipeline was built using the NetApp DataOps Toolkit, AFF storage, and NVIDIA DGX System. The pipeline performs batch audio signal processing, automatic speech recognition (ASR), transfer learning, and sentiment analysis leveraging the DataOps Toolkit, NVIDIA Riva SDK, and the Tao framework. Expanding the sentiment analysis use case to the financial services industry, we built a SparkNLP workflow, loaded three BERT models for various NLP tasks, such as named entity recognition, and obtained sentence-level sentiment for NASDAQ Top 10 companies’ quarterly earnings calls.

The following script sentiment_analysis_spark.py uses the FinBERT model to process transcripts in
HDFS and produce positive, neutral, and negative sentiment counts, as shown in the following table:

```
-bash-4.2$ time ~/anaconda3/bin/spark-submit
--packages com.johnsnowlabs.nlp:spark-nlp_2.12:3.4.3
--master yarn
--executor-memory 5g
--executor-cores 1
--num-executors 160
--conf spark.driver.extraJavaOptions="-Xss10m -XX:MaxPermSize=1024M"
--conf spark.executor.extraJavaOptions="-Xss10m -XX:MaxPermSize=512M"
/sparkusecase/tr-4570-nlp/sentiment_analysis_spark.py
hdfs:///data1/Transcripts/
> ./sentiment_analysis_hdfs.log 2>&1
real13m14.300s
user557m11.319s
sys4m47.676s
```

The following table lists the earnings-call, sentence-level sentiment analysis for NASDAQ Top 10 companies from 2016 to 2020.

<table>
<thead>
<tr>
<th>Sentiment counts and percentage</th>
<th>All 10 Companies</th>
<th>AAPL</th>
<th>AMD</th>
<th>AMZN</th>
<th>CSCO</th>
<th>GOOGL</th>
<th>INTC</th>
<th>MSFT</th>
<th>NVDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive counts</td>
<td>7447</td>
<td>1567</td>
<td>743</td>
<td>290</td>
<td>682</td>
<td>826</td>
<td>824</td>
<td>904</td>
<td>417</td>
</tr>
<tr>
<td>Neutral counts</td>
<td>64067</td>
<td>6856</td>
<td>7596</td>
<td>5086</td>
<td>6650</td>
<td>5914</td>
<td>6099</td>
<td>5715</td>
<td>6189</td>
</tr>
<tr>
<td>Negative counts</td>
<td>1787</td>
<td>253</td>
<td>213</td>
<td>84</td>
<td>189</td>
<td>97</td>
<td>282</td>
<td>202</td>
<td>89</td>
</tr>
<tr>
<td>Uncategorized counts</td>
<td>196</td>
<td>0</td>
<td>0</td>
<td>76</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(total counts)</td>
<td>73497</td>
<td>8676</td>
<td>8552</td>
<td>5536</td>
<td>7521</td>
<td>6837</td>
<td>7205</td>
<td>6822</td>
<td>6695</td>
</tr>
</tbody>
</table>

In terms of percentages, most sentences spoken by the CEOs and CFOs are factual and therefore carry neutral sentiment. During an earnings call, analysts ask questions which might convey positive or negative sentiment. It is worth further investigating quantitatively how negative or positive sentiment affect stock prices on the same or next day of trading.

The following table lists the sentence-level sentiment analysis for NASDAQ Top 10 companies, expressed in percentage.
In terms of the workflow runtime, we saw a significant 4.78x improvement from *local* mode to a distributed environment in HDFS, and a further 0.14% improvement by leveraging NFS.

As the following figure shows, data and model parallelism improved the data processing and distributed TensorFlow model inferencing speed. Data location in NFS yielded a slightly better runtime because the workflow bottleneck is the downloading of pretrained models. If we increase the transcripts dataset size, the advantage of NFS is more obvious.
Distributed training with Horovod performance

The following command produced runtime information and a log file in our Spark cluster using a single master node with 160 executors each with one core. The executor memory was limited to 5GB to avoid out-of-memory error. See the section “Python scripts for each major use case” for more detail regarding the data processing, model training, and model accuracy calculation in keras_spark_horovod_rossmann_estimator.py.

```
(base) [root@n138 horovod]# time spark-submit
--master local
--executor-memory 5g
--executor-cores 1
--num-executors 160
/sparkusecase/horovod/keras_spark_horovod_rossmann_estimator.py
--epochs 10
--data-dir file:///sparkusecase/horovod
--local-submission-csv /tmp/submission_0.csv
--local-checkpoint-file /tmp/checkpoint/
> /tmp/keras_spark_horovod_rossmann_estimator_local. log 2>&1
```

The resulting runtime with ten training epochs was as follows:

```
real 43m34.608s
user 12m22.057s
sys 2m30.127s
```

It took more than 43 minutes to process input data, train a DNN model, calculate accuracy, and produce
TensorFlow checkpoints and a CSV file for prediction results. We limited the number of training epochs to 10, which in practice is often set to 100 to ensure satisfactory model accuracy. The training time typically scales linearly with the number of epochs.

We next used the four worker nodes available in the cluster and executed the same script in yarn mode with data in HDFS:

```
(base) [root@n138 horovod]# time spark-submit
    --master yarn
    --executor-memory 5g
    --executor-cores 1 --num-executors 160
    /sparkusecase/horovod/keras_spark_horovod_rossmann_estimator.py
    --epochs 10
    --data-dir hdfs:///user/hdfs/tr-4570/experiments/horovod
    --local-submission-csv /tmp/submission_1.csv
    --local-checkpoint-file /tmp/checkpoint/
    > /tmp/keras_spark_horovod_rossmann_estimator_yarn.log 2>&1
```

The resulting runtime was improved as follows:

```
real8m13.728s
user7m48.421s
sys1m26.063s
```

With Horovod’s model and data parallelism in Spark, we saw a 5.29x runtime speedup of yarn versus local mode with ten training epochs. This is shown in the following figure with the legends HDFS and Local. The underlying TensorFlow DNN model training can be further accelerated with GPUs if available. We plan to conduct this testing and publish results in a future technical report.

Our next test compared the runtimes with input data residing in NFS versus HDFS. The NFS volume on the AFF A800 was mounted on /sparkdemo/horovod across the five nodes (one master, four workers) in our Spark cluster. We ran a similar command as for previous tests, with the `--data-dir` parameter now pointing to the NFS mount:

```
(base) [root@n138 horovod]# time spark-submit
    --master yarn
    --executor-memory 5g
    --executor-cores 1
    --num-executors 160
    /sparkusecase/horovod/keras_spark_horovod_rossmann_estimator.py
    --epochs 10
    --data-dir file:///sparkdemo/horovod
    --local-submission-csv /tmp/submission_2.csv
    --local-checkpoint-file /tmp/checkpoint/
    > /tmp/keras_spark_horovod_rossmann_estimator_nfs.log 2>&1
```
The resulting runtime with NFS was as follows:

```
real 5m46.229s  
user 5m35.693s  
sys  1m5.615s  
```

There was a further 1.43x speedup, as shown in the following figure. Therefore, with a NetApp all-flash storage connected to their cluster, customers enjoy the benefits of fast data transfer and distribution for Horovod Spark workflows, achieving 7.55x speedup versus running on a single node.

![Horovod Spark Workflow Runtime](image)

Deep learning models for CTR prediction performance

For recommender systems designed to maximize CTR, you must learn sophisticated feature interactions behind user behaviors that can be mathematically calculated from low order to high order. Both low-order and high-order feature interactions should be equally important for a good deep learning model without biasing towards one or the other. Deep Factorization Machine (DeepFM), a factorization machine-based neural network, combines factorization machines for recommendation and deep learning for feature learning in a new neural network architecture.

Although conventional factorization machines model pairwise feature interactions as an inner product of latent vectors between features and can theoretically capture high-order information, in practice, machine learning practitioners usually only use second-order feature interactions due to the high computation and storage complexity. Deep neural network variants like Google’s **Wide & Deep Models** on the other hand learns sophisticated feature interactions in a hybrid network structure by combining a linear wide model and a deep model.

There are two inputs to this **Wide & Deep Model**, one for the underlying wide model and the other for the deep, the latter part of which still requires expert feature engineering and thus renders the technique less generalizable to other domains. Unlike the **Wide & Deep Model**, DeepFM can be efficiently trained with raw features without any feature engineering because its wide part and deep part share the same input and the
We first processed the Criteo train.txt (11GB) file into a CSV file named ctr_train.csv stored in an NFS mount /sparkdemo/tr-4570-data using run_classification_criteo_spark.py from the section “Python scripts for each major use case.” Within this script, the function process_input_file performs several string methods to remove tabs and insert ‘,’ as the delimiter and ‘\n’ as newline. Note that you only need to process the original train.txt once, so that the code block is shown as comments.

For the following testing of different DL models, we used ctr_train.csv as the input file. In subsequent testing runs, the input CSV file was read into a Spark DataFrame with schema containing a field of ‘label’, integer dense features ['I1', 'I2', 'I3', …, 'I13'], and sparse features ['C1', 'C2', 'C3', …, 'C26']. The following spark-submit command takes in an input CSV, trains DeepFM models with 20% split for cross validation, and picks the best model after ten training epochs to calculate prediction accuracy on the testing set:

```
(spark) [root@n138 ~]# time spark-submit --master yarn --executor-memory 5g --executor-cores 1 --num-executors 160 /sparkusecase/DeepCTR/examples/run_classification_criteo_spark.py --data-dir file:///sparkdemo/tr-4570-data > /tmp/run_classification_criteo_spark_local.log 2>&1
```

Note that since the data file ctr_train.csv is over 11GB, you must set a sufficient spark.driver.maxResultSize greater than the dataset size to avoid error.

```
spark = SparkSession.builder \
  .master("yarn") \n  .appName("deep_ctr_classification") \n  .config("spark.jars.packages", "io.github.ravwojdyla:spark-schema-utils_2.12:0.1.0") \n  .config("spark.executor.cores", "1") \n  .config("spark.executor.memory", "5gb") \n  .config("spark.executor.memoryOverhead", "1500") \n  .config("spark.driver.memoryOverhead", "1500") \n  .config("spark.sql.shuffle.partitions", "480") \n  .config("spark.sql.execution.arrow.enabled", "true") \n  .config("spark.driver.maxResultSize", "50gb") \n  .getOrCreate()
```

In the above SparkSession.builder configuration we also enabled Apache Arrow, which converts a Spark DataFrame into a Pandas DataFrame with the df.toPandas() method.

```
22/06/17 15:56:21 INFO scheduler.DAGScheduler: Job 2 finished: toPandas at /sparkusecase/DeepCTR/examples/run_classification_criteo_spark.py:96, took 627.126487 s
Obtained Spark DF and transformed to Pandas DF using Arrow.
```
After random splitting, there are over 36M rows in the training dataset and 9M samples in the testing set:

Training dataset size = 36672493
Testing dataset size = 9168124

Because this technical report is focused on CPU testing without using any GPUs, it is imperative that you build TensorFlow with appropriate compiler flags. This step avoids invoking any GPU-accelerated libraries and takes full advantage of TensorFlow's Advanced Vector Extensions (AVX) and AVX2 instructions. These features are designed for linear algebraic computations like vectorized addition, matrix multiplications inside a feed-forward, or back-propagation DNN training. Fused Multiply Add (FMA) instruction available with AVX2 using 256-bit floating point (FP) registers is ideal for integer code and data types, resulting in up to a 2x speedup. For FP code and data types, AVX2 achieves 8% speedup over AVX.

To build TensorFlow from source, NetApp recommends using Bazel. For our environment, we executed the following commands in the shell prompt to install dnf, dnf-plugins, and Bazel:

```
yum install dnf
dnf install 'dnf-command(copr)'
dnf copr enable vbatts/bazel
dnf install bazel5
```

You must enable GCC 5 or newer to use C++17 features during the build process, which is provided by RHEL with Software Collections Library (SCL). The following commands install devtoolset and GCC 11.2.1 on our RHEL 7.9 cluster:

```
subscription-manager repos --enable rhel-server-rhscl-7-rpms
yum install devtoolset-11-toolchain
yum install devtoolset-11-gcc-c++
yum update
scl enable devtoolset-11 bash
  . /opt/rh/devtoolset-11/enable
```

Note that the last two commands enable devtoolset-11, which uses /opt/rh/devtoolset-11/root/usr/bin/gcc (GCC 11.2.1). Also, make sure your git version is greater than 1.8.3 (this comes with RHEL 7.9). Refer to this article for updating git to 2.24.1.

We assume that you have already cloned the latest TensorFlow master repo. Then create a workspace
directory with a WORKSPACE file to build TensorFlow from source with AVX, AVX2, and FMA. Run the configure file and specify the correct Python binary location. CUDA is disabled for our testing because we did not use a GPU. A .bazelrc file is generated according to your settings. Further, we edited the file and set build --define=no_hdfs_support=false to enable HDFS support. Refer to .bazelrc in the section “Python scripts for each major use case,” for a complete list of settings and flags.

```bash
./configure
bazel build -c opt --copt=-mavx --copt=-mavx2 --copt=-mfma --copt=
-mfpmath=both -k //tensorflow/tools/pip_package:build_pip_package
```

After you build TensorFlow with the correct flags, run the following script to process the Criteo Display Ads dataset, train a DeepFM model, and calculate the Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores.

```bash
(base) [root@nl38 examples]# ~/anaconda3/bin/spark-submit
--master yarn
--executor-memory 15g
--executor-cores 1
--num-executors 160
/sparkusecase/DeepCTR/examples/run_classification_criteo_spark.py
--data-dir file:///sparkdemo/tr-4570-data
> . /run_classification_criteo_spark_nfs.log 2>&1
```

After ten training epochs, we obtained the AUC score on the testing dataset:
In a manner similar to previous use cases, we compared the Spark workflow runtime with data residing in different locations. The following figure shows a comparison of the deep learning CTR prediction for a Spark workflows runtime.
Hybrid cloud solution

A modern enterprise data center is a hybrid cloud that connects multiple distributed infrastructure environments through a continuous data management plane with a consistent operating model, on premises and/or in multiple public clouds. To get the most out of a hybrid cloud, you must be able to seamlessly move data between your on-premises and multi-cloud environments without the need for any data conversions or application refactoring.

Customers have indicated that they start their hybrid cloud journey either by moving secondary storage to the cloud for use cases such as data protection or by moving less business-critical workloads such as application development and DevOps to the cloud. They then move on to more critical workloads. Web and content hosting, DevOps and application development, databases, analytics, and containerized apps are among the most popular hybrid-cloud workloads. The complexity, cost, and risks of enterprise AI projects have historically hindered AI adoption from experimental stage to production.

With a NetApp hybrid-cloud solution, customers benefit from integrated security, data governance, and compliance tools with a single control panel for data and workflow management across distributed environments, while optimizing the total cost of ownership based on their consumption. The following figure is an example solution of a cloud service partner tasked with providing multi-cloud connectivity for customers' big-data-analytics data.
In this scenario, IoT data received in AWS from different sources is stored in a central location in NetApp Private Storage (NPS). The NPS storage is connected to Spark or Hadoop clusters located in AWS and Azure enabling big-data-analytics applications running in multiple clouds accessing the same data. The main requirements and challenges for this use case include the following:

- Customers want to run analytics jobs on the same data using multiple clouds.
- Data must be received from different sources such as on-premises and cloud environments through different sensors and hubs.
- The solution must be efficient and cost effective.
- The main challenge is to build a cost-effective and efficient solution that delivers hybrid analytics services between different on-premises and cloud environments.

Our data protection and multicloud connectivity solution resolves the pain point of having cloud analytics applications across multiple hyperscalers. As shown in the figure above, data from sensors is streamed and ingested into the AWS Spark cluster through Kafka. The data is stored in an NFS share residing in NPS, which is located outside of the cloud provider within an Equinix data center.

Because NetApp NPS is connected to Amazon AWS and Microsoft Azure through Direct Connect and Express Route connections respectively, customers can leverage the In-Place Analytics Module to access the data from both Amazon and AWS analytics clusters. Consequently, because both on-premises and NPS storage runs ONTAP software, SnapMirror can mirror the NPS data into the on-premises cluster, providing hybrid cloud analytics across on-premises and multiple clouds.

For the best performance, NetApp typically recommends using multiple network interfaces and direct connection or express routes to access the data from cloud instances. We have other data mover solutions including XCP and BlueXP Copy and Sync to help customers build application-aware, secure, and cost-effective hybrid-cloud Spark clusters.

**Python scripts for each major use case**

The following three Python scripts correspond to the three major use cases tested. First is `sentiment_analysis_sparknlp.py`.

```python
# TR-4570 Refresh NLP testing by Rick Huang
from sys import argv
```
import os
import sparknlp
import pyspark.sql.functions as F
from sparknlp import Finisher
from pyspark.ml import Pipeline
from sparknlp.base import *
from sparknlp.annotator import *
from sparknlp.pretrained import PretrainedPipeline
from sparknlp import Finisher
# Start Spark Session with Spark NLP
spark = sparknlp.start()
print("Spark NLP version:")
print(sparknlp.version())
print("Apache Spark version:")
print(spark.version)
spark = sparknlp.SparkSession.builder \
    .master("yarn") \n    .appName("test_hdfs_read_write") \n    .config("spark.executor.cores", "1") \n    .config("spark.jars.packages", "com.johnsnowlabs.nlp:spark-nlp_2.12:3.4.3")\n    .config('spark.executor.memory', '5gb') \n    .config('spark.executor.memoryOverhead', '1000')\n    .config('spark.driver.memoryOverhead', '1000')\n    .config("spark.sql.shuffle.partitions", "480")\n    .getOrCreate()
sc = spark.sparkContext
from pyspark.sql import SQLContext
sql = SQLContext(sc)
sqlContext = SQLContext(sc)
# Download pre-trained pipelines & sequence classifier
explain_pipeline_model = PretrainedPipeline('explain_document_dl', lang='en').model#pipeline_sa = PretrainedPipeline("classifierdl_bertwiki_finance_sentiment_pipeline", lang="en")
# pipeline_finbert = BertForSequenceClassification.loadSavedModel('/sparkusecase/bert_sequence_classifier_finbert_en_3', spark)
sequenceClassifier = BertForSequenceClassification
    .pretrained('bert_sequence_classifier_finbert', 'en') \n    .setInputCols(["token", "document"])
    .setOutputCol('class') \n    .setCaseSensitive(True) \n    .setMaxSentenceLength(512)
def process_sentence_df(data):
    # Pre-process: begin

1644
print("1. Begin DataFrame pre-processing...\n")
print(f"\n\t2. Attaching DocumentAssembler Transformer to the pipeline")
documentAssembler = DocumentAssembler() \  
  .setInputCol("text") \  
  .setOutputCol("document") \  
  .setCleanupMode("inplace_full")  
# .setCleanupMode("shrink", "inplace_full")
doc_df = documentAssembler.transform(data)
doc_df.printSchema()
doc_df.show(truncate=50)

# Pre-process: get rid of blank lines
clean_df = doc_df.withColumn("tmp", F.explode("document")) \  
  .select("tmp.result").where("tmp.end != -1") .withColumnRenamed("result", "text").dropna()
print("[OK!] DataFrame after initial cleanup:\n")
clean_df.printSchema()
clean_df.show(truncate=80)

# for FinBERT
tokenizer = Tokenizer() \  
  .setInputCols(['document']) \  
  .setOutputCol('token')
print(f"\n\t3. Attaching Tokenizer Annotator to the pipeline")

pipeline_finbert = Pipeline(stages=[
  documentAssembler,
  tokenizer,
  sequenceClassifier
])

# Use Finisher() & construct PySpark ML pipeline
finisher = Finisher().setInputCols(["token", "lemma", "pos", "entities"])

print(f"\n\t4. Attaching Finisher Transformer to the pipeline")

pipeline_ex = Pipeline() \  
  .setStages([  
    explain_pipeline_model,
    finisher
  ])  
print("\n\t\t ---- Pipeline Built Successfully ----")

# Loading pipelines to annotate
#result_ex_df = pipeline_ex.transform(clean_df)
ex_model = pipeline_ex.fit(clean_df)
annotations_finished_ex_df = ex_model.transform(clean_df)
# result_sa_df = pipeline_sa.transform(clean_df)
result_finbert_df = pipeline_finbert.fit(clean_df).transform(clean_df)
print("\n\t\t ---- Document Explain, Sentiment Analysis & FinBERT Pipeline Fitted Successfully ----")
# Check the result entities
print("[OK!] Simple explain ML pipeline result:
")
annotations_finished_ex_df.printSchema()
annotations_finished_ex_df.select('text', 'finished_entities').show(truncate=False)
# Check the result sentiment from FinBERT
print("[OK!] Sentiment Analysis FinBERT pipeline result:
")
result_finbert_df.printSchema()
result_finbert_df.select('text', 'class.result').show(80, False)
sentiment_stats(result_finbert_df)
return
def sentiment_stats(finbert_df):
    result_df = finbert_df.select('text', 'class.result')
    sa_df = result_df.select('result')
    sa_df.groupBy('result').count().show()
    # total_lines = result_clean_df.count()
    # num_neutral = result_clean_df.where(result_clean_df.result == ['neutral']).count()
    # num_positive = result_clean_df.where(result_clean_df.result == ['positive']).count()
    # num_negative = result_clean_df.where(result_clean_df.result == ['negative']).count()
    # print(f"\nRatio of neutral sentiment = {num_neutral/total_lines}\n")
    # print(f"Ratio of positive sentiment = {num_positive / total_lines}\n")
    # print(f"Ratio of negative sentiment = {num_negative / total_lines}\n")
    return

def process_input_file(file_name):
    # Turn input file to Spark DataFrame
    print("START processing input file...")
data_df = spark.read.text(file_name)
data_df.show()
    # rename first column 'text' for sparknlp
output_df = data_df.withColumnRenamed("value", "text").dropna()
output_df.printSchema()
return output_df
def process_local_dir(directory):
    filelist = []
    for subdir, dirs, files in os.walk(directory):
        for filename in files:
            filepath = subdir + os.sep + filename
            print("[OK!] Will process the following files:")
            if filepath.endswith(".txt"):
                print(filepath)
                filelist.append(filepath)
    return filelist
def process_local_dir_or_file(dir_or_file):
if os.path.isfile(dir_or_file):
    input_df = process_input_file(dir_or_file)
    print("Obtained input_df.")
    process_sentence_df(input_df)
    print("Processed input_df")
    numfiles += 1
else:
    filelist = process_local_dir(dir_or_file)
    for file in filelist:
        input_df = process_input_file(file)
        process_sentence_df(input_df)
        numfiles += 1
    return numfiles

def process_hdfs_dir(dir_name):
    # Turn input files to Spark DataFrame
    print("START processing input HDFS directory...")
    data_df = spark.read.option("recursiveFileLookup", "true").text(dir_name)
    data_df.show()
    print("[DEBUG] total lines in data_df = ", data_df.count())
    # rename first column 'text' for sparknlp
    output_df = data_df.withColumnRenamed("value", "text").dropna()
    output_df.show(40, False)
    print("[DEBUG] HDFS dir resulting data_df schema: 
")
    output_df.printSchema()
    process_sentence_df(output_df)
    print("Processed HDFS directory: ", dir_name)

if __name__ == '__main__':
    try:
        if len(argv) == 2:
            print("Start processing input...
")
    except:
        print("[ERROR] Please enter input text file or path to process!\n")
        exit(1)
    # This is for local file, not hdfs:
    numfiles = process_local_dir_or_file(str(argv[1]))
    # For HDFS single file & directory:
    input_df = process_input_file(str(argv[1]))
    print("Obtained input_df.")
    process_sentence_df(input_df)
    print("Processed input_df")
    numfiles += 1
    # For HDFS directory of subdirectories of files:
The second script is `keras_spark_horovod_rossmann_estimator.py`.
if args.master:
    conf.setMaster(args.master)
elif args.num_proc:
    conf.setMaster('local[{}].format(args.num_proc))
spark = SparkSession.builder.config(conf=conf).getOrCreate()
train_csv = spark.read.csv('{}train.csv' % args.data_dir,
header=True)
test_csv = spark.read.csv('{}test.csv' % args.data_dir, header=True)
store_csv = spark.read.csv('{}store.csv' % args.data_dir,
header=True)
store_states_csv = spark.read.csv('{}store_states.csv' %
args.data_dir, header=True)
state_names_csv = spark.read.csv('{}state_names.csv' % args.data_dir,
header=True)
googletrend_csv = spark.read.csv('{}googletrend.csv' %
args.data_dir, header=True)
weather_csv = spark.read.csv('{}weather.csv' % args.data_dir,
header=True)

def expand_date(df):
    df = df.withColumn('Date', df.Date.cast(T.DateType()))
    return df
        .withColumn('Year', F.year(df.Date))
        .withColumn('Month', F.month(df.Date))
        .withColumn('Week', F.weekofyear(df.Date))
        .withColumn('Day', F.dayofmonth(df.Date))

def prepare_google_trend():
    # Extract week start date and state.
    google_trend_all = googletrend_csv
        .withColumn('Date', F.regexp_extract(googletrend_csv.week,
'(.*) -', 1))
        .withColumn('State', F.regexp_extract(googletrend_csv.file,
'Rossmann_DE_(.*)', 1))
    # Map state NI -> HB,NI to align with other data sources.
    google_trend_all = google_trend_all
        .withColumn('State', F.when(google_trend_all.State == 'NI',
'HB,NI').otherwise(google_trend_all.State))
    # Expand dates.
    return expand_date(google_trend_all)

def add_elapsed(df, cols):
    def add_elapsed_column(col, asc):
        def fn(rows):
            last_store, last_date = None, None
            for row in rows:
                store, date = row[col.split('.')]
                if asc:
                    if store > last_store and date > last_date:
                        return True
                else:
                    if store < last_store and date < last_date:
                        return True
                last_store, last_date = store, date
            return False
        return fn
for r in rows:
    if last_store != r.Store:
        last_store = r.Store
        last_date = r.Date
    if r[col]:
        last_date = r.Date
    fields = r.asDict().copy()
    fields[('After' if asc else 'Before') + col] = (r.Date
- last_date).days
    yield Row(**fields)
return fn
df = df.repartition(df.Store)
for asc in [False, True]:
    sort_col = df.Date.asc() if asc else df.Date.desc()
    rdd = df.sortWithinPartitions(df.Store.asc(), sort_col).rdd
    for col in cols:
        rdd = rdd.mapPartitions(add_elapsed_column(col, asc))
    df = rdd.toDF()
return df
def prepare_df(df):
    num_rows = df.count()
    # Expand dates.
    df = expand_date(df)
    df = df |
        .withColumn('Open', df.Open != '0') |
        .withColumn('Promo', df.Promo != '0') |
        .withColumn('StateHoliday', df.StateHoliday != '0') |
        .withColumn('SchoolHoliday', df.SchoolHoliday != '0')
    # Merge in store information.
    store = store_csv.join(store_states_csv, 'Store')
    df = df.join(store, 'Store')
    # Merge in Google Trend information.
    google_trend_all = prepare_google_trend()
    df = df.join(google_trend_all, ['State', 'Year', 'Week']).select(df['*'], google_trend_all.trend)
    # Merge in Google Trend for whole Germany.
    google_trend_de = google_trend_all[google_trend_all.file == 'Rossmann_DE'].withColumnRenamed('trend', 'trend_de')
    df = df.join(google_trend_de, ['Year', 'Week']).select(df['*'], google_trend_de.trend_de)
    # Merge in weather.
    weather = weather_csv.join(state_names_csv, weather_csv.file == state_names_csv.StateName)
    df = df.join(weather, ['State', 'Date'])
    # Fix null values.
    df = df \
withColumn('CompetitionOpenSinceYear',
    F.coalesce(df.CompetitionOpenSinceYear, F.lit(1900))) \n     .withColumn('CompetitionOpenSinceMonth',
    F.coalesce(df.CompetitionOpenSinceMonth, F.lit(1))) \n     .withColumn('Promo2SinceYear', F.coalesce(df.Promo2SinceYear, F.lit(1900))) \n     .withColumn('Promo2SinceWeek', F.coalesce(df.Promo2SinceWeek, F.lit(1)))
# Days & months competition was open, cap to 2 years.
df = df.withColumn('CompetitionOpenSince',
    F.to_date(F.format_string('%s-%s-15',
    df.CompetitionOpenSinceYear,
    df.CompetitionOpenSinceMonth)))
df = df.withColumn('CompetitionOpenSinceYear',
    F.when(df.CompetitionOpenSinceYear > 1900,
    F.greatest(F.lit(0), F.least(F.lit(360 * 2), F.datediff(df.Date, df.CompetitionOpenSince))))
    .otherwise(0))
df = df.withColumn('CompetitionDaysOpen',
    (df.CompetitionDaysOpen / 30).cast(T.IntegerType()))
# Days & weeks of promotion, cap to 25 weeks.
df = df.withColumn('Promo2Since',
    F.expr('date_add(format_string("%s-01-01",
    Promo2SinceYear), (cast(Promo2SinceWeek as int) - 1) * 7)'))
df = df.withColumn('Promo2Days',
    F.when(df.Promo2SinceYear > 1900,
    F.greatest(F.lit(0), F.least(F.lit(25 * 7), F.datediff(df.Date, df.Promo2Since))))
    .otherwise(0))
df = df.withColumn('Promo2Weeks', (df.Promo2Days / 7).cast(T.IntegerType()))
# Check that we did not lose any rows through inner joins.
assert num_rows == df.count(), 'lost rows in joins'
return df

def build_vocabulary(df, cols):
    vocab = {}
    for col in cols:
        values = [r[0] for r in df.select(col).distinct().collect()]
        col_type = type([x for x in values if x is not None][0])
        default_value = col_type()
        vocab[col] = sorted(values, key=lambda x: x or default_value)
    return vocab
def cast_columns(df, cols):
    for col in cols:
        df = df.withColumn(col,
F.coalesce(df[col].cast(T.FloatType()), F.lit(0.0)))

return df

def lookup_columns(df, vocab):
    def lookup(mapping):
        def fn(v):
            return mapping.index(v)
        return F.udf(fn, returnType=T.IntegerType())
    for col, mapping in vocab.items():
        df = df.withColumn(col, lookup(mapping)(df[col]))
    return df

if args.sample_rate:
    train_csv = train_csv.sample(withReplacement=False, fraction=args.sample_rate)
    test_csv = test_csv.sample(withReplacement=False, fraction=args.sample_rate)

    # Prepare data frames from CSV files.
    train_df = prepare_df(train_csv).cache()
    test_df = prepare_df(test_csv).cache()

    # Add elapsed times from holidays & promos, the data spanning training & test datasets.
    elapsed_cols = ['Promo', 'StateHoliday', 'SchoolHoliday']
    elapsed = add_elapsed(train_df.select('Date', 'Store', *elapsed_cols)
                          .unionAll(test_df.select('Date', 'Store', *
                                        *elapsed_cols)),
                          elapsed_cols)

    # Join with elapsed times.
    train_df = train_df .join(elapsed, ['Date', 'Store']) \\ .select(train_df['*'], *[prefix + col for prefix in ['Before', 'After'] for col in elapsed_cols])
    test_df = test_df .join(elapsed, ['Date', 'Store']) \\ .select(test_df['*'], *[prefix + col for prefix in ['Before', 'After'] for col in elapsed_cols])

    # Filter out zero sales.
    train_df = train_df.filter(train_df.Sales > 0)

print('===============')
print('Prepared data frame')
print('===============
train_df.show()
categorical_cols = [
    'Store', 'State', 'DayOfWeek', 'Year', 'Month', 'Day', 'Week', 'CompetitionMonthsOpen', 'Promo2Weeks', 'StoreType', 'Assortment', 'PromoInterval', 'CompetitionOpenSinceYear', 'Promo2SinceYear', 'Events', 'Promo', 'StateHoliday', 'SchoolHoliday']
continuous_cols = [
    'CompetitionDistance', 'Max_TemperatureC', 'Mean_TemperatureC', 'Min_TemperatureC', 'Max_Humidity',
    'Mean_Humidity', 'Min_Humidity', 'Max_Wind_SpeedKm_h', 'Mean_Wind_SpeedKm_h', 'CloudCover', 'trend', 'trend_de',
    'BeforePromo', 'AfterPromo', 'AfterStateHoliday', 'BeforeStateHoliday', 'BeforeSchoolHoliday', 'AfterSchoolHoliday'
]
all_cols = categorical_cols + continuous_cols
# Select features.
train_df = train_df.select(*(all_cols + ['Sales', 'Date'])).cache()
test_df = test_df.select(*(all_cols + ['Id', 'Date'])).cache()
# Build vocabulary of categorical columns.
vocab = build_vocabulary(train_df.select(*categorical_cols)
    .unionAll(test_df.select(*categorical_cols)).cache(),
categorical_cols)
# Cast continuous columns to float & lookup categorical columns.
train_df = cast_columns(train_df, continuous_cols + ['Sales'])
train_df = lookup_columns(train_df, vocab)
test_df = cast_columns(test_df, continuous_cols)
test_df = lookup_columns(test_df, vocab)
# Split into training & validation.
# Test set is in 2015, use the same period in 2014 from the training
# set as a validation set.
test_min_date = test_df.agg(F.min(test_df.Date)).collect()[0][0]
test_max_date = test_df.agg(F.max(test_df.Date)).collect()[0][0]
one_year = datetime.timedelta(365)
train_df = train_df.withColumn('Validation',
    (train_df.Date > test_min_date -
    one_year) & (train_df.Date <= test_max_date -
    one_year))
# Determine max Sales number.
max_sales = train_df.agg(F.max(train_df.Sales)).collect()[0][0]
# Convert Sales to log domain
train_df = train_df.withColumn('Sales', F.log(train_df.Sales))
print('----------------------------------')
print('Data frame with transformed columns')
print('----------------------------------')
train_df.show()
print('----------------')
print('Data frame sizes')
print('----------------')
train_rows = train_df.filter(~train_df.Validation).count()
val_rows = train_df.filter(train_df.Validation).count()
test_rows = test_df.count()
print('Training: %d' % train_rows)
print('Validation: %d' % val_rows)
print('Test: %d' % test_rows)
# ============== #
# MODEL TRAINING #
# ============== #
print('==============')
print('Model training')
print('==============')
def exp_rmspe(y_true, y_pred):
    """Competition evaluation metric, expects logarithmic inputs."""
    pct = tf.square((tf.exp(y_true) - tf.exp(y_pred)) /
        tf.exp(y_true))
    # Compute mean excluding stores with zero denominator.
    x = tf.reduce_sum(tf.where(y_true > 0.001, pct,
        tf.zeros_like(pct)))
    y = tf.reduce_sum(tf.where(y_true > 0.001, tf.ones_like(pct),
        tf.zeros_like(pct)))
    return tf.sqrt(x / y)
def act_sigmoid_scaled(x):
    """Sigmoid scaled to logarithm of maximum sales scaled by 20%."""
    return tf.nn.sigmoid(x) * tf.math.log(max_sales) * 1.2

CUSTOM_OBJECTS = {'exp_rmspe': exp_rmspe,
    'act_sigmoid_scaled': act_sigmoid_scaled}

# Disable GPUs when building the model to prevent memory leaks
if LooseVersion(tf.__version__) >= LooseVersion('2.0.0'):
    # See https://github.com/tensorflow/tensorflow/issues/33168
    os.environ['CUDA_VISIBLE_DEVICES'] = '-1'
else:

    K.set_session(tf.Session(config=tf.ConfigProto(device_count={'GPU': 0})))
    # Build the model.
    inputs = {col: Input(shape=(1,), name=col) for col in all_cols}
    embeddings = [Embedding(len(vocab[col]), 10, input_length=1,
        name='emb_' + col)(inputs[col])
        for col in categorical_cols]
    continuous_bn = Concatenate()([Reshape((1, 1), name='reshape_' +
        col)(inputs[col])
        for col in continuous_cols])
    continuous_bn = BatchNormalization()(continuous_bn)
    x = Concatenate()(embeddings + [continuous_bn])
    x = Flatten()(x)
    x = Dense(1000, activation='relu',
        kernel_regularizer=tf.keras.regularizers.l2(0.00005))(x)
    x = Dense(1000, activation='relu',
        kernel_regularizer=tf.keras.regularizers.l2(0.00005))(x)
x = Dense(1000, activation='relu',
           kernel_regularizer=tf.keras.regularizers.l2(0.00005))(x)
x = Dense(500, activation='relu',
           kernel_regularizer=tf.keras.regularizers.l2(0.00005))(x)
x = Dropout(0.5)(x)
output = Dense(1, activation=act_sigmoid_scaled)(x)
model = tf.keras.Model(inputs[all_cols], output)
model.summary()
opt = tf.keras.optimizers.Adam(lr=args.learning_rate, epsilon=1e-3)
# Checkpoint callback to specify options for the returned Keras model
ckpt_callback = BestModelCheckpoint(monitor='val_loss', mode='auto',
                                      save_freq='epoch')
# Horovod: run training.
store = Store.create(args.work_dir)
backend = SparkBackend(num_proc=args.num_proc,
                        stdout=sys.stdout, stderr=sys.stderr,
                        prefix_output_with_timestamp=True)
keras_estimator = hvd.KerasEstimator(backend=backend,
                                      store=store,
                                      model=model,
                                      optimizer=opt,
                                      loss='mae',
                                      metrics=[exp_rmspe],
                                      custom_objects=CUSTOM_OBJECTS,
                                      feature_cols=all_cols,
                                      label_cols=['Sales'],
                                      validation='Validation',
                                      batch_size=args.batch_size,
                                      epochs=args.epochs,
                                      verbose=2,
                                      checkpoint_callback=ckpt_callback)
kmodel =
keras_estimator.fit(train_df).setOutputCols(['Sales_output'])
history = keras_model.getHistory()
best_val_rmspe = min(history['val_exp_rmspe'])
print('Best RMSPE: %f' % best_val_rmspe)
# Save the trained model.
kmodel.save(args.local_checkpoint_file)
print('Written checkpoint to %s' % args.local_checkpoint_file)
# ================ #
# FINAL PREDICTION #
# ================ #
print('------------------------------
print('Final prediction')
print('------------------------------')
pred_df=keras_model.transform(test_df)
pred_df.printSchema()
pred_df.show(5)
# Convert from log domain to real Sales numbers
pred_df=pred_df.withColumn('Sales_pred', F.exp(pred_df.Sales_output))
submission_df = pred_df.select(pred_df.Id.cast(T.IntegerType()),
pred_df.Sales_pred).toPandas()
    submission_df.sort_values(by=['Id']).to_csv(args.local_submission_csv,
index=False)
    print('Saved predictions to %s' % args.local_submission_csv)
spark.stop()

The third script is run_classification_criteo_spark.py.

import tempfile, string, random, os, uuid
import argparse, datetime, sys, shutil
import csv
import numpy as np
from sklearn.model_selection import train_test_split
from tensorflow.keras.callbacks import EarlyStopping
from pyspark import SparkContext
from pyspark.sql import SparkSession, SQLContext, Row, DataFrame
from pyspark.mllib import linalg as mllib_linalg
from pyspark.mllib.linalg import SparseVector as mllibSparseVector
from pyspark.mllib.linalg import VectorUDT as mllibVectorUDT
from pyspark.mllib.linalg import Vector as mllibVector, Vectors as mllibVectors
from pyspark.mllib.regression import LabeledPoint
from pyspark.mllib.classification import LogisticRegressionWithSGD
from pyspark.ml import linalg as ml_linalg
from pyspark.ml.linalg import VectorUDT as mlVectorUDT
from pyspark.ml.linalg import SparseVector as mlSparseVector
from pyspark.ml.linalg import Vector as mlVector, Vectors as mlVectors
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.feature import OneHotEncoder
from math import log
from math import exp  # exp(-t) = e^-t
from operator import add
from pyspark.sql.functions import udf, split, lit
from pyspark.sql.functions import size, sum as sqlsum
import pyspark.sql.functions as F
import pyspark.sql.types as T
import pyspark.sql.functions as explode, col, log, when
from collections import defaultdict
import pandas as pd
import pyspark.pandas as ps
from sklearn.metrics import log_loss, roc_auc_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
from deepctr.models import DeepFM
from deepctr.feature_column import SparseFeat, DenseFeat,
get_feature_names
spark = SparkSession.builder \\
    .master("yarn") \\
    .appName("deep_ctr_classification") \\
    .config("spark.jars.packages", "io.github.ravwojdyla:spark-schema-utils_2.12:0.1.0") \\
    .config("spark.executor.cores", "1") \\
    .config('spark.executor.memory', '5gb') \\
    .config('spark.executor.memoryOverhead', '1500') \\
    .config('spark.driver.memoryOverhead', '1500') \\
    .config('spark.sql.shuffle.partitions", "480") \\
    .config("spark.sql.execution.arrow.enabled", "true") \\
    .config("spark.driver.maxResultSize", "50gb") \\
    .getOrCreate()
# spark.conf.set("spark.sql.execution.arrow.enabled", "true") # deprecated
print("Apache Spark version:")
print(spark.version)
sc = spark.sparkContext
sqlContext = SQLContext(sc)
parser = argparse.ArgumentParser(description='Spark DCN CTR Prediction
Example',
formatter_class=argparse.ArgumentDefaultsHelpFormatter)
parser.add_argument('--data-dir', default='file://' + os.getcwd(),
help='location of data on local filesystem (prefixed
with file://) or on HDFS')
def process_input_file(file_name, sparse_feat, dense_feat):
    # Need this preprocessing to turn Criteo raw file into CSV:
    print("START processing input file..."
    # only convert the file ONCE
    # sample = open(file_name)
    # sample = '
'.join([str(x.replace('
', '').replace('	', ',')) for x in sample])
    # Add header in data file and save as CSV
    # header = ','.join(str(x) for x in ([label'] + dense_feat +
    sparse_feat))
    # with open('/sparkdemo/tr-4570-data/ctr_train.csv', mode='w',
    encoding="utf-8") as f:
```python
# f.write(header + '\n' + sample)
# f.close()
print("Raw training file processed and saved as CSV: ", f.name)
raw_df = sqlContext.read.option("header", True).csv(file_name)
raw_df.show(5, False)
raw_df.printSchema()
# convert columns I1 to I13 from string to integers
conv_df = raw_df.select(col('label').cast("double"),
    *(col(i).cast("float").alias(i) for i in raw_df.columns if i in dense_feat),
    *(col(c) for c in raw_df.columns if c in sparse_feat))
print("Schema of raw_df with integer columns type changed:")
conv_df.printSchema()
# result_pdf = conv_df.select("*").toPandas()
tmp_df = conv_df.na.fill(0, dense_feat)
result_df = tmp_df.na.fill('-1', sparse_feat)
result_df.show()
return result_df
if __name__ == "__main__":
    args = parser.parse_args()
    # Pandas read CSV
    # data = pd.read_csv('%s/criteo_sample.txt' % args.data_dir)
    # print("Obtained Pandas df.")
    dense_features = ["I" + str(i) for i in range(1, 14)]
    sparse_features = ["C" + str(i) for i in range(1, 27)]
    # Spark read CSV
    # process_input_file('%s/train.txt' % args.data_dir, sparse_features, dense_features) # run only ONCE
    spark_df = process_input_file('%s/data.txt' % args.data_dir, sparse_features, dense_features) # sample data
    # spark_df = process_input_file('%s/ctr_train.csv' % args.data_dir, sparse_features, dense_features)
    print("Obtained Spark df and filled in missing features.")
    data = spark_df
    # Pandas
    #data[sparse_features] = data[sparse_features].fillna('-1', )
    #data[dense_features] = data[dense_features].fillna(0, )
    target = ['label']
    label_npa = data.select("label").toPandas().to_numpy()
    print("label numPy array has length = ", len(label_npa)) # 45,840,617 w/ 11GB dataset
    label_npa.ravel()
    label_npa.reshape(len(label_npa), )
    # 1.Label Encoding for sparse features,and do simple Transformation for dense features
```
print("Before LabelEncoder():")
data.printSchema()  # label: float (nullable = true)
for feat in sparse_features:
    lbe = LabelEncoder()
    tmp_pdf = data.select(feat).toPandas().to_numpy()
    tmp_ndarray = lbe.fit_transform(tmp_pdf)
    print("After LabelEncoder(), tmp_ndarray[0] =", tmp_ndarray[0])
    # print("Data tmp PDF after lbe transformation, the output ndarray
    has length = ", len(tmp Ndarray))  # 45,840,617 for 11GB dataset
    tmp Ndarray.ravel()
    tmp Ndarray.reshape(len(tmp Ndarray), )
    out Ndarray = np.column_stack([label Npa, tmp Ndarray])
    pdf = pd.DataFrame(out Ndarray, columns=['label', feat])
    s_df = spark.createDataFrame(pdf)
    s_df.printSchema()  # label: double (nullable = true)
    print("Before joining data df with s_df, s_df example rows:")
    s_df.show(1, False)
    data = data.drop(feat).join(s_df, 'label').drop('label')
    print("After LabelEncoder(), data df example rows:")
    data.show(1, False)
    print("Finished processing sparse_features: ", feat)
    data.show()
    data.printSchema()
    mms = MinMaxScaler(feature_range=(0, 1))
    # data[dense_features] = mms.fit_transform(data[dense_features]) # for
    # Pandas df
    tmp_pdf = data.select(dense_features).toPandas().to_numpy()
    tmp_ndarray = mms.fit_transform(tmp_pdf)
    tmp Ndarray.ravel()
    tmp Ndarray.reshape(len(tmp Ndarray), len(tmp Ndarray[0]))
    out Ndarray = np.column_stack([label Npa, tmp Ndarray])
    pdf = pd.DataFrame(out Ndarray, columns=['label'] + dense_features)
    s_df = spark.createDataFrame(pdf)
    s_df.printSchema()
    data.drop(*dense_features).join(s_df, 'label').drop('label')
    print("Finished processing dense_features: ", dense_features)
    data.show()

    # 2.count #unique features for each sparse field, and record dense
    feature field name
    fixlen_feature_columns = [SparseFeat(feat,
        vocabulary_size=data.select(feat).distinct().count() + 1, embedding_dim=4)
        for i, feat in enumerate(sparse_features)] +
In this document, we discuss the Apache Spark architecture, customer use cases, and the NetApp storage portfolio as it relates to big data, modern analytics, and AI, ML, and
DL. In our performance validation tests based on industry-standard benchmarking tools and customer demand, the NetApp Spark solutions demonstrated superior performance relative to native Hadoop systems. A combination of the customer use cases and performance results presented in this report can help you to choose an appropriate Spark solution for your deployment.

Where to find additional information

The following references were used in this TR:

- Apache Spark architecture and components
  http://spark.apache.org/docs/latest/cluster-overview.html
- Apache Spark use cases
  https://www.qubole.com/blog/big-data/apache-spark-use-cases/
- Apache challenges
- Spark NLP
  https://www.johnsnowlabs.com/spark-nlp/
- BERT
  https://arxiv.org/abs/1810.04805
- Deep and Cross Network for Ad Click Predictions
  https://arxiv.org/abs/1708.05123
- FlexGroup
- Streaming ETL
  https://www.infoq.com/articles/apache-spark-streaming
- NetApp E-Series Solutions for Hadoop
  https://www.netapp.com/media/16420-tr-3969.pdf
- Sentiment Analysis from Customer Communications with NetApp AI
- NetApp Modern Data Analytics Solutions
Big Data Analytics Data to Artificial Intelligence

TR-4732: Big data analytics data to artificial intelligence

Karthikeyan Nagalingam, NetApp

This document describes how to move big-data analytics data and HPC data to AI. AI processes NFS data through NFS exports, whereas customers often have their AI data in a big-data analytics platform, such as HDFS, Blob, or S3 storage as well as HPC platforms such as GPFS. This paper provides guidelines for moving big-data-analytics data and HPC data to AI by using NetApp XCP and NIPAM. We also discuss the business benefits of moving data from big data and HPC to AI.

Concepts and components

Big data analytics storage

Big data analytics is the major storage provider for HDFS. A customer often uses a Hadoop-compatible file system (HCFS) such as Windows Azure Blob Storage, MapR File System (MapR-FS), and S3 object storage.

General parallel file system

IBM’s GPFS is an enterprise file system that provides an alternative to HDFS. GPFS provides flexibility for applications to decide the block size and replication layout, which provide good performance and efficiency.

NetApp In-Place Analytics Module

The NetApp In-Place Analytics Module (NIPAM) serves as a driver for Hadoop clusters to access NFS data. It has four components: a connection pool, an NFS InputStream, a file handle cache, and an NFS OutputStream. For more information, see TR-4382: NetApp In-Place Analytics Module.

Hadoop Distributed Copy

Hadoop Distributed Copy (DistCp) is a distributed copy tool used for large inter-cluster and intra-cluster coping tasks. This tool uses MapReduce for data distribution, error handling, and reporting. It expands the list of files and directories and inputs them to map tasks to copy the data from the source list. The image below shows the
Hadoop DistCp moves data between the two HDFS systems without using an additional driver. NetApp provides the driver for non-HDFS systems. For an NFS destination, NIPAM provides the driver to copy data that Hadoop DistCp uses to communicate with NFS destinations when copying data.

**NetApp Cloud Volumes Service**

The NetApp Cloud Volumes Service is a cloud-native file service with extreme performance. This service helps customers accelerate their time-to-market by rapidly spinning resources up and down and using NetApp features to improve productivity and reduce staff downtime. The Cloud Volumes Service is the right alternative for disaster recovery and back up to cloud because it reduces the overall data-center footprint and consumes less native public cloud storage.

**NetApp XCP**

NetApp XCP is client software that enables fast and reliable any-to-NetApp and NetApp-to-NetApp data migration. This tool is designed to copy a large amount of unstructured NAS data from any NAS system to a NetApp storage controller. The XCP Migration Tool uses a multicore, multichannel I/O streaming engine that can process many requests in parallel, such as data migration, file or directory listings, and space reporting. This is the default NetApp data Migration Tool. You can use XCP to copy data from a Hadoop cluster and HPC to NetApp NFS storage. The diagram below shows data transfer from a Hadoop and HPC cluster to a NetApp NFS volume using XCP.
NetApp BlueXP Copy and Sync

NetApp BlueXP Copy and Sync is a hybrid data replication software-as-a-service that transfers and synchronizes NFS, S3, and CIFS data seamlessly and securely between on-premises storage and cloud storage. This software is used for data migration, archiving, collaboration, analytics, and more. After data is transferred, BlueXP Copy and Sync continuously syncs the data between the source and destination. Going forward, it then transfers the delta. It also secures the data within your own network, in the cloud, or on premises. This software is based on a pay-as-you-go model, which provides a cost-effective solution and provides monitoring and reporting capabilities for your data transfer.

Customer challenges

Customers might face the following challenges when trying to access data from big-data analytics for AI operations:

- Customer data is in a data lake repository. The data lake can contain different types of data such as structured, unstructured, semi-structured, logs, and machine-to-machine data. All these data types must be processed in AI systems.
- AI is not compatible with Hadoop file systems. A typical AI architecture is not able to directly access HDFS and HCFS data, which must be moved to an AI-understandable file system (NFS).
- Moving data lake data to AI typically requires specialized processes. The amount of data in the data lake can be very large. A customer must have an efficient, high-throughput, and cost-effective way to move data into AI systems.
- Syncing data. If a customer wants to sync data between the big-data platform and AI, sometimes the data processed through AI can be used with big data for analytical processing.

Data mover solution

In a big-data cluster, data is stored in HDFS or HCFS, such as MapR-FS, the Windows Azure Storage Blob, S3, or the Google file system. We performed testing with HDFS, MapR-FS, and S3 as the source to copy data to NetApp ONTAP NFS export with the help of NIPAM by using the hadoop distcp command from the source.
The following diagram illustrates the typical data movement from a Spark cluster running with HDFS storage to a NetApp ONTAP NFS volume so that NVIDIA can process AI operations.

The `hadoop distcp` command uses the MapReduce program to copy the data. NIPAM works with MapReduce to act as a driver for the Hadoop cluster when copying data. NIPAM can distribute a load across multiple network interfaces for a single export. This process maximizes the network throughput by distributing the data across multiple network interfaces when you copy the data from HDFS or HCFS to NFS.

![Diagram](image)

NIPAM is not supported or certified with MapR.

**Data mover solution for AI**

The data mover solution for AI is based on customers' needs to process Hadoop data from AI operations. NetApp moves data from HDFS to NFS by using the NIPAM. In one use case, the customer needed to move data to NFS on the premises and another customer needed to move data from the Windows Azure Storage Blob to Cloud Volumes Service in order to process the data from the GPU cloud instances in the cloud.

The following diagram illustrates the data mover solution details.
The following steps are required to build the data mover solution:

1. ONTAP SAN provides HDFS, and NAS provides the NFS volume through NIPAM to the production data lake cluster.

2. The customer's data is in HDFS and NFS. The NFS data can be production data from other applications that is used for big data analytics and AI operations.

3. NetApp FlexClone technology creates a clone of the production NFS volume and provisions it to the AI cluster on premises.

4. Data from an HDFS SAN LUN is copied into an NFS volume with NIPAM and the `hadoop distcp` command. NIPAM uses the bandwidth of multiple network interfaces to transfer data. This process reduces the data copy time so that more data can be transferred.

5. Both NFS volumes are provisioned to the AI cluster for AI operations.

6. To process on-the-premises NFS data with GPUs in the cloud, the NFS volumes are mirrored to NetApp Private Storage (NPS) with NetApp SnapMirror technology and mounted to cloud service providers for GPUs.

7. The customer wants to process data in EC2/EMR, HDInsight, or DataProc services in GPUs from cloud service providers. The Hadoop data mover moves the data from Hadoop services to the Cloud Volumes Services with NIPAM and the `hadoop distcp` command.

8. The Cloud Volumes Service data is provisioned to AI through the NFS protocol. Data that is processed through AI can be sent on an on-premises location for big data analytics in addition to the NVIDIA cluster through NIPAM, SnapMirror, and NPS.

In this scenario, the customer has large file-count data in the NAS system at a remote location that is required for AI processing on the NetApp storage controller on premises. In this scenario, it’s better to use the XCP Migration Tool to migrate the data at a faster speed.

The hybrid-use-case customer can use BlueXP Copy and Sync to migrate on-premises data from NFS, CIFS, and S3 data to the cloud and vice versa for AI processing by using GPUs such as those in an NVIDIA cluster. Both BlueXP Copy and Sync and the XCP Migration Tool are used for the NFS data migration to NetApp ONTAP NFS.
In this validation, we used four servers as Network Shared Disk (NSD) servers to provide physical disks for GPFS. GPFS is created on top of the NSD disks to export them as NFS exports so that NFS clients can access them, as shown in the figure below. We used XCP to copy the data from GPFS- exported NFS to a NetApp NFS volume.
GPFS essentials

The following node types are used in GPFS:

- **Admin node.** Specifies an optional field containing a node name used by the administration commands to communicate between nodes. For example, the admin node `mastr-51.netapp.com` could pass a network check to all other nodes in the cluster.

- **Quorum node.** Determines whether a node is included in the pool of nodes from which quorum is derived. You need at least one node as a quorum node.

- **Manager Node.** Indicates whether a node is part of the node pool from which file system managers and token managers can be selected. It is a good idea to define more than one node as a manager node. How many nodes you designate as manager depends on the workload and the number of GPFS server licenses you have. If you are running large parallel jobs, you might need more manager nodes than in a four-node cluster supporting a web application.

- **NSD Server.** The server that prepares each physical disk for use with GPFS.

- **Protocol node.** The node that shares GPFS data directly through any Secure Shell (SSH) protocol with the NFS. This node requires a GPFS server license.

List of operations for GPFS, NFS, and XCP

This section provides the list of operations that create GPFS, export GPFS as an NFS export, and transfer the data by using XCP.

Create GPFS

To create GPFS, complete the following steps:

1. Download and install spectrum-scale data access for the Linux version on one of the servers.
2. Install the prerequisite package (chef for example) in all nodes and disable Security-Enhanced Linux (SELinux) in all nodes.
3. Set up the install node and add the admin node and the GPFS node to the cluster definition file.
4. Add the manager node, the quorum node, the NSD servers, and the GPFS node.
5. Add the GUI, admin, and GPFS nodes, and add an additional GUI server if required.
6. Add another GPFS node and check the list of all nodes.
7. Specify a cluster name, profile, remote shell binary, remote file copy binary, and port range to be set on all the GPFS nodes in the cluster definition file.
8. View the GPFS configuration settings and add an additional admin node.
9. Disable the data collection and upload the data package to the IBM Support Center.
10. Enable NTP and precheck the configurations before install.
11. Configure, create, and check the NSD disks.
12. Create the GPFS.
13. Mount the GPFS.
14. Verify and provide the required permissions to the GPFS.
15. Verify the GPFS read and write by running the `dd` command.
Export GPFS into NFS

To export the GPFS into NFS, complete the following steps:

1. Export GPFS as NFS through the `/etc/exports` file.
2. Install the required NFS server packages.
3. Start the NFS service.
4. List the files in the GPFS to validate the NFS client.

Configure NFS client

To configure the NFS client, complete the following steps:

1. Export the GPFS as NFS through the `/etc/exports` file.
2. Start the NFS client services.
3. Mount the GPFS through the NFS protocol on the NFS client.
4. Validate the list of GPFS files in the NFS mounted folder.
5. Move the data from GPFS exported NFS to NetApp NFS by using XCP.
6. Validate the GPFS files on the NFS client.

HDFS and MapR-FS to ONTAP NFS

For this solution, NetApp validated the migration of data from data lake (HDFS) and MapR cluster data to ONTAP NFS. The data resided in MapR-FS and HDFS. NetApp XCP introduced a new feature that directly migrates the data from a distributed file system such as HDFS and MapR-FS to ONTAP NFS. XCP uses async threads and HDFS C API calls to communicate and transfer data from MapR-FS as well as HDFS.

The below figure shows the data migration from data lake (HDFS) and MapR-FS to ONTAP NFS. With this new feature, you don’t have to export the source as an NFS share.

Why are customers moving from HDFS and MapR-FS to NFS?

Most of the Hadoop distributions such as Cloudera and Hortonworks use HDFS and MapR distributions uses
their own filesystem called Mapr-FS to store data. HDFS and MapR-FS data provides the valuable insights to
data scientists that can be leveraged in machine learning (ML) and deep learning (DL). The data in HDFS and
MapR-FS is not shared, which means it cannot be used by other applications. Customers are looking for
shared data, specifically in the banking sector where customers’ sensitive data is used by multiple applications.
The latest version of Hadoop (3.x or later) supports NFS data source, which can be accessed without
additional third-party software. With the new NetApp XCP feature, data can be moved directly from HDFS and
MapR-FS to NetApp NFS in order to provide access to multiple applications

Testing was done in Amazon Web Services (AWS) to transfer the data from MapR-FS to NFS for the initial
performance test with 12 MAPR nodes and 4 NFS servers.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Size</th>
<th>vCPU</th>
<th>Memory</th>
<th>Storage</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFS server</td>
<td>4</td>
<td>i3en.24xlarge</td>
<td>96</td>
<td>488GiB</td>
<td>8x 7500 NVMe SSD</td>
</tr>
<tr>
<td>MapR nodes</td>
<td>12</td>
<td>I3en.12xlarge</td>
<td>48</td>
<td>384GiB</td>
<td>4x 7500 NVMe SSD</td>
</tr>
</tbody>
</table>

Based on initial testing, we obtained 20GBps throughput and were able to transfer 2PB per day of data.

For more information about HDFS data migration without exporting HDFS to NFS, see the “Deployment steps -
NAS” section in TR-4863: TR-4863: Best-Practice Guidelines for NetApp XCP - Data Mover, File Migration,
and Analytics.

**Business benefits**

Moving data from big data analytics to AI provides the following benefits:

- The ability to extract data from different Hadoop file systems and GPFS into a unified NFS storage system
- A Hadoop-integrated and automated way to transfer data
- A reduction in the cost of library development for moving data from Hadoop file systems
- Maximum performance by aggregated throughput of multiple network interfaces from a single source of
data by using NIPAM
- Scheduled and on-demand methods to transfer data
- Storage efficiency and enterprise management capability for unified NFS data by using ONTAP data
management software
- Zero cost for data movement with the Hadoop method for data transfer

**GPFS to NFS-Detailed steps**

This section provides the detailed steps needed to configure GPFS and move data into
NFS by using NetApp XCP.

**Configure GPFS**

1. Download and Install Spectrum Scale Data Access for Linux on one of the servers.
2. Install the prerequisite package (including chef and the kernel headers) on all nodes.

[root@mastr-51 5.0.3.1]# for i in 51 53 136 138 140 ; do ssh 10.63.150.$i "hostname; rpm -ivh /gpfs_install/chef* "; done
mastr-51.netapp.com
warning: /gpfs_install/chef-13.6.4-1.el7.x86_64.rpm: Header V4 DSA/SHA1 Signature, key ID 83ef826a: NOKEY
Preparing...
package chef-13.6.4-1.el7.x86_64 is already installed
mastr-53.netapp.com
warning: /gpfs_install/chef-13.6.4-1.el7.x86_64.rpm: Header V4 DSA/SHA1 Signature, key ID 83ef826a: NOKEY
Preparing...
Updating / installing...
chef-13.6.4-1.el7
Thank you for installing Chef!
workr-136.netapp.com
warning: /gpfs_install/chef-13.6.4-1.el7.x86_64.rpm: Header V4 DSA/SHA1 Signature, key ID 83ef826a: NOKEY
Preparing...
Updating / installing...
chef-13.6.4-1.el7
Thank you for installing Chef!
workr-138.netapp.com
warning: /gpfs_install/chef-13.6.4-1.el7.x86_64.rpm: Header V4 DSA/SHA1 Signature, key ID 83ef826a: NOKEY
3. Disable SELinux in all nodes.

```
[root@mastr-51 5.0.3.1]# for i in 51 53 136 138 140 ; do ssh 10.63.150.$i "hostname; sudo setenforce 0"; done
mastr-51.netapp.com
setenforce: SELinux is disabled
mastr-53.netapp.com
setenforce: SELinux is disabled
workr-136.netapp.com
setenforce: SELinux is disabled
workr-138.netapp.com
setenforce: SELinux is disabled
workr-140.netapp.com
setenforce: SELinux is disabled
[root@mastr-51 5.0.3.1]#
```

4. Set up the install node.
5. Add the admin node and the GPFS node to the cluster definition file.

```
[root@mastr-51 installer]# ./spectrumscale node add mastr-51 -a
[ INFO ] Adding node mastr-51.netapp.com as a GPFS node.
[ INFO ] Setting mastr-51.netapp.com as an admin node.
[ INFO ] Configuration updated.
[ INFO ] Tip: Designate protocol, nsd and admin nodes in your environment to use during install: ./spectrumscale -v node add <node> -p-a-n
[root@mastr-51 installer]#
```

6. Add the manager node and the GPFS node.

```
[root@mastr-51 installer]# ./spectrumscale node add mastr-53 -m
[ INFO ] Adding node mastr-53.netapp.com as a GPFS node.
[ INFO ] Adding node mastr-53.netapp.com as a manager node.
[root@mastr-51 installer]#
```

7. Add the quorum node and the GPFS node.

```
[root@mastr-51 installer]# ./spectrumscale node add workr-136 -q
[ INFO ] Adding node workr-136.netapp.com as a GPFS node.
[ INFO ] Adding node workr-136.netapp.com as a quorum node.
[root@mastr-51 installer]#
```

8. Add the NSD servers and the GPFS node.
9. Add the GUI, admin, and GPFS nodes.

```
[root@mastr-51 installer]# ./spectrumscale node add workr-136 -g
[root@mastr-51 installer]# ./spectrumscale node add workr-136 -a
[ INFO ] Setting workr-136.netapp.com as an admin node.
[ INFO ] Configuration updated.
[ INFO ] Tip: Designate protocol or nsd nodes in your environment to use during install: ./spectrumscale node add <node> -p -n
[root@mastr-51 installer]#
```

10. Add another GUI server.

```
[root@mastr-51 installer]# ./spectrumscale node add mastr-53 -g
[root@mastr-51 installer]#
```

11. Add another GPFS node.

```
[root@mastr-51 installer]# ./spectrumscale node add workr-140
[ INFO ] Adding node workr-140.netapp.com as a GPFS node.
[root@mastr-51 installer]#
```

12. Verify and list all nodes.
[root@mastr-51 installer]# ./spectrumscale node list
[ INFO ] List of nodes in current configuration:
[ INFO ] [Installer Node]
[ INFO ] 10.63.150.51
[ INFO ] [Cluster Details]
[ INFO ] No cluster name configured
[ INFO ] Setup Type: Spectrum Scale
[ INFO ] [Extended Features]
[ INFO ] File Audit logging : Disabled
[ INFO ] Watch folder : Disabled
[ INFO ] Management GUI : Enabled
[ INFO ] Performance Monitoring : Disabled
[ INFO ] Callhome : Enabled
[ INFO ] GPFS GUI Callhome OS Arch
Admin Quorum Manager NSD Protocol
[ INFO ] Node Server Node Node Server Node
Node
[ INFO ] mastr-51.netapp.com X
rhe17 x86_64
[ INFO ] mastr-53.netapp.com X
X rhe17 x86_64
[ INFO ] workr-136.netapp.com X X
X rhe17 x86_64
[ INFO ] workr-138.netapp.com X
rhe17 x86_64
[ INFO ] workr-140.netapp.com
rhe17 x86_64
[ INFO ] [Export IP address]
[ INFO ] No export IP addresses configured
[root@mastr-51 installer]#

13. Specify a cluster name in the cluster definition file.

[root@mastr-51 installer]# ./spectrumscale config gpfs -c mastr-51.netapp.com
[ INFO ] Setting GPFS cluster name to mastr-51.netapp.com
[root@mastr-51 installer]#

14. Specify the profile.
15. Specify the remote shell binary to be used by GPFS; use `-r` argument.

   [root@mastr-51 installer]# ./spectrumscale config gpfs -r /usr/bin/ssh
   [ INFO ] Setting Remote shell command to /usr/bin/ssh

16. Specify the remote file copy binary to be used by GPFS; use `-rc` argument.

   [root@mastr-51 installer]# ./spectrumscale config gpfs -rc /usr/bin/scp
   [ INFO ] Setting Remote file copy command to /usr/bin/scp

17. Specify the port range to be set on all GPFS nodes; use `-e` argument.

   [root@mastr-51 installer]# ./spectrumscale config gpfs -e 60000-65000
   [ INFO ] Setting GPFS Daemon communication port range to 60000-65000

18. View the GPFS config settings.

   [root@mastr-51 installer]# ./spectrumscale config gpfs --list
   [ INFO ] Current settings are as follows:
   [ INFO ] GPFS profile is default.
   [ INFO ] Remote shell command is /usr/bin/ssh.
   [ INFO ] Remote file copy command is /usr/bin/scp.
   [ INFO ] GPFS Daemon communication port range is 60000-65000.

19. Add an admin node.
20. Disable the data collection and upload the data package to the IBM Support Center.

21. Enable NTP.
22. Precheck the configurations before Install.
23. Configure the NSD disks.
24. Create the NSD disks.

[root@mastr-51 cluster-test]# mmcrnsd -F disk.1st -v no
mmcrnsd: Processing disk sdf
mmcrnsd: Processing disk sdf
mmcrnsd: Propagating the cluster configuration data to all
    affected nodes. This is an asynchronous process.
[root@mastr-51 cluster-test]#

25. Check the NSD disk status.

[root@mastr-51 cluster-test]# mmlsnsd

File system   Disk name    NSD servers
-----------------------------------------------
    (free disk)   nsd1         workr-136.netapp.com
    (free disk)   nsd2         workr-138.netapp.com

[root@mastr-51 cluster-test]#

26. Create the GPFS.
The following disks of gpfs1 will be formatted on node workr-136.netapp.com:
  nsd1: size 3814912 MB
  nsd2: size 3814912 MB
Formatting file system ...
Disks up to size 33.12 TB can be added to storage pool system.
Creating Inode File
Creating Allocation Maps
Creating Log Files
Clearing Inode Allocation Map
Clearing Block Allocation Map
Formatting Allocation Map for storage pool system
Completed creation of file system /dev/gpfs1.
mmcrrfs: Propagating the cluster configuration data to all affected nodes. This is an asynchronous process.

27. Mount the GPFS.

28. Check and provide the required permissions to the GPFS.
29. Check the GPFS read and write by running the `dd` command.

```bash
[root@mastr-51 cluster-test]# for i in 51 53 136 138 ; do ssh 10.63.150.$i "hostname; ls -ltrh /gpfs1" ; done;
mastr-51.netapp.com
mastr-53.netapp.com
workr-136.netapp.com
workr-138.netapp.com

[root@mastr-51 cluster-test]#
```

```bash
[root@mastr-51 cluster-test]# dd if=/dev/zero of=/gpfs1/testfile bs=1024M count=5
5+0 records in
5+0 records out
5368709120 bytes (5.4 GB) copied, 8.3981 s, 639 MB/s
```

```bash
[root@mastr-51 cluster-test]# for i in 51 53 136 138 ; do ssh 10.63.150.$i "hostname; ls -ltrh /gpfs1" ; done;
mastr-51.netapp.com
mastr-53.netapp.com
workr-136.netapp.com
workr-138.netapp.com
```

```
[root@mastr-51 cluster-test]# dd if=/dev/zero of=/gpfs1/testfile bs=1024M count=5
5+0 records in
5+0 records out
5368709120 bytes (5.4 GB) copied, 8.3981 s, 639 MB/s
```

```
[root@mastr-51 cluster-test]# for i in 51 53 136 138 ; do ssh 10.63.150.$i "hostname; ls -ltrh /gpfs1" ; done;
mastr-51.netapp.com
mastr-53.netapp.com
workr-136.netapp.com
workr-138.netapp.com
```

```
```
Export GPFS into NFS

To export GPFS into NFS, complete the following steps:

1. Export the GPFS as NFS through the `/etc/exports` file.

```
[root@mastr-51 gpfs1]# cat /etc/exports
/gpfs1        *(rw,fsid=745)
[root@mastr-51 gpfs1]
```

2. Install the required NFS server packages.

```
[root@mastr-51 ~]# yum install rpcbind
Loaded plugins: priorities, product-id, search-disabled-repos, subscription-manager
Resolving Dependencies
--> Running transaction check
--- Package rpcbind.x86_64 0:0.2.0-47.el7 will be updated
--- Package rpcbind.x86_64 0:0.2.0-48.el7 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

========================================================================
-- Package                                               Arch
--- Version                                                    Repository
--- Size
--- Reporting 1 Package
--- x86_64
--- rhel-7-server-rpms
--- 60 k

Transaction Summary

========================================================================
--- Upgrade  1 Package
```
Total download size: 60 k
Is this ok [y/d/N]: y
Downloading packages:
No Presto metadata available for rhel-7-server-rpms
rpcbind-0.2.0-48.el7.x86_64.rpm
|  60 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating : rpcbind-0.2.0-48.el7.x86_64
1/2
  Cleanup   : rpcbind-0.2.0-47.el7.x86_64
2/2
  Verifying : rpcbind-0.2.0-48.el7.x86_64
1/2
  Verifying : rpcbind-0.2.0-47.el7.x86_64
2/2
Updated:
  rpcbind.x86_64 0:0.2.0-48.el7

Complete!
[root@mastr-51 ~]#

3. Start the NFS service.
4. List the files in GPFS to validate the NFS client.
<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Type</th>
<th>Size</th>
<th>Used</th>
<th>Avail</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/mapper/rhel_stlrx300s6--22--irmc-root</td>
<td>xfs</td>
<td>94G</td>
<td>55G</td>
<td>39G</td>
</tr>
<tr>
<td>Use% Mounted on</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/devtmpfs</td>
<td>devtmpfs</td>
<td>32G</td>
<td>0</td>
<td>32G</td>
</tr>
<tr>
<td>0% /dev</td>
<td>tmpfs</td>
<td>32G</td>
<td>0</td>
<td>32G</td>
</tr>
<tr>
<td>0% /dev/shm</td>
<td>tmpfs</td>
<td>32G</td>
<td>3.3G</td>
<td>29G</td>
</tr>
<tr>
<td>0% /sys/fs/cgroup</td>
<td>tmpfs</td>
<td>32G</td>
<td>0</td>
<td>32G</td>
</tr>
<tr>
<td>/dev/sda7</td>
<td>xfs</td>
<td>9.4G</td>
<td>210M</td>
<td>9.1G</td>
</tr>
<tr>
<td>3% /boot</td>
<td>tmpfs</td>
<td>6.3G</td>
<td>0</td>
<td>6.3G</td>
</tr>
<tr>
<td>0% /run/user/10065</td>
<td>tmpfs</td>
<td>6.3G</td>
<td>0</td>
<td>6.3G</td>
</tr>
<tr>
<td>0% /run/user/10068</td>
<td>tmpfs</td>
<td>6.3G</td>
<td>0</td>
<td>6.3G</td>
</tr>
<tr>
<td>0% /run/user/10069</td>
<td>tmpfs</td>
<td>6.3G</td>
<td>0</td>
<td>6.3G</td>
</tr>
<tr>
<td>10.63.150.213:/nc_volume3</td>
<td>nfs4</td>
<td>380G</td>
<td>8.0M</td>
<td>380G</td>
</tr>
<tr>
<td>1% /mnt</td>
<td>tmpfs</td>
<td>6.3G</td>
<td>0</td>
<td>6.3G</td>
</tr>
<tr>
<td>0% /run/user/0</td>
<td>gpfs</td>
<td>7.3T</td>
<td>9.1G</td>
<td>7.3T</td>
</tr>
<tr>
<td>1% /gpfs1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
[root@mastr-51 gpfs1]# df -Th
Filesystem Type  Size  Used  Avail Use% Mounted on
/dev/mapper/rhel_stlrx300s6--22--irmc-root xfs  94G   55G   39G  59% /
devtmpfs devtmpfs 32G   0    32G   0% /dev
tmpfs tmpfs 32G   0    32G   0% /dev/shm
tmpfs tmpfs 32G   3.3G  29G   11% /run
tmpfs tmpfs 32G   0    32G   0% /sys/fs/cgroup
dev/sda7 xfs 9.4G  210M  9.1G  3% /boot
tmpfs tmpfs 6.3G  0    6.3G  0% /run/user/10065
tmpfs tmpfs 6.3G  0    6.3G  0% /run/user/10068
tmpfs tmpfs 6.3G  0    6.3G  0% /run/user/10069
10.63.150.213:/nc_volume3 nfs4 380G  8.0M 380G 1% /mnt
tmpfs tmpfs 6.3G  0    6.3G  0% /run/user/0
gpfs1 gpfs 7.3T  9.1G 7.3T 1% /gpfs1
```

```
[root@mastr-51 gpfs1]# cd /gpfs1
[root@mastr-51 ~]# ls
catalog ces gpfs-ces ha testfile
```

```
[root@mastr-51 ~]# cd /gpfs1
[root@mastr-51 gpfs1]# ls
catalog ces gpfs-ces ha testfile
```

```
[root@mastr-51 gpfs1]# ls -ltr ha
total 5.1G
```

```
```
```
```
Configure the NFS client

To configure the NFS client, complete the following steps:

1. Install packages in the NFS client.

   ```
   [root@hdp2 ~]# yum install nfs-utils rpcbind
   Loaded plugins: product-id, search-disabled-repos, subscription-manager
   HDP-2.6-GPL-repo-4
   |  2.9 kB  00:00:00
   HDP-2.6-repo-4
   |  2.9 kB  00:00:00
   HDP-3.0-GPL-repo-2
   |  2.9 kB  00:00:00
   HDP-3.0-repo-2
   |  2.9 kB  00:00:00
   HDP-3.0-repo-3
   |  2.9 kB  00:00:00
   HDP-3.1-repo-1
   |  2.9 kB  00:00:00
   HDP-3.1-repo-51
   |  2.9 kB  00:00:00
   HDP-UTILS-1.1.0.22-repo-1
   |  2.9 kB  00:00:00
   HDP-UTILS-1.1.0.22-repo-2
   |  2.9 kB  00:00:00
   HDP-UTILS-1.1.0.22-repo-3
   |  2.9 kB  00:00:00
   HDP-UTILS-1.1.0.22-repo-4
   |  2.9 kB  00:00:00
   HDP-UTILS-1.1.0.22-repo-51
   |  2.9 kB  00:00:00
   ambari-2.7.3.0
   |  2.9 kB  00:00:00
   epel/x86_64/metalink
   |  13 kB  00:00:00
   epel
   |  5.3 kB  00:00:00
   mysql-connectors-community
   |  2.5 kB  00:00:00
   mysql-tools-community
   |  2.5 kB  00:00:00
   mysql56-community
   |  2.5 kB  00:00:00
   rhel-7-server-optional-rpms
   |  3.2 kB  00:00:00
   rhel-7-server-rpms
   ```
Resolving Dependencies
--- Running transaction check
---- Package nfs-utils.x86_64 1:1.3.0-0.61.el7 will be updated
---- Package nfs-utils.x86_64 1:1.3.0-0.65.el7 will be an update
---- Package rpcbind.x86_64 0:0.2.0-47.el7 will be updated
---- Package rpcbind.x86_64 0:0.2.0-48.el7 will be an update
--- Finished Dependency Resolution

Dependencies Resolved

========================================================================
<table>
<thead>
<tr>
<th>Package</th>
<th>Arch</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository</td>
<td>Size</td>
<td></td>
</tr>
</tbody>
</table>
|---------------------------------------------------------------
| Updating: | | |
| nfs-utils | x86_64 | 1:1.3.0-0.65.el7 |
| rhel-7-server-rpms | 412 k |
| rpcbind | x86_64 | 0.2.0-48.el7 |
| rhel-7-server-rpms | 60 k |

Transaction Summary
========================================================================

========================================================================
Upgrade 2 Packages

Total download size: 472 k
Is this ok [y/d/N]: y
Downloading packages:
No Presto metadata available for rhel-7-server-rpms
(1/2): rpcbind-0.2.0-48.el7.x86_64.rpm
| 60 kB  00:00:00
(2/2): nfs-utils-1.3.0-0.65.el7.x86_64.rpm
| 412 kB  00:00:00
---------------------------------------------------------------
Total
1.2 MB/s | 472 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating   : rpcbind-0.2.0-48.el7.x86_64
1/4
  service rpcbind start
  Updating   : 1:nfs-utils-1.3.0-0.65.el7.x86_64
2/4
  Cleanup    : 1:nfs-utils-1.3.0-0.61.el7.x86_64
3/4
  Cleanup    : rpcbind-0.2.0-47.el7.x86_64
4/4
  Verifying  : 1:nfs-utils-1.3.0-0.65.el7.x86_64
1/4
  Verifying  : rpcbind-0.2.0-48.el7.x86_64
2/4
  Verifying  : rpcbind-0.2.0-47.el7.x86_64
3/4
  Verifying  : 1:nfs-utils-1.3.0-0.61.el7.x86_64
4/4
Updated:
  nfs-utils.x86_64 1:1.3.0-0.65.el7
  rpcbind.x86_64 0:0.2.0-48.el7

Complete!
[root@hdp2 ~]#

2. Start the NFS client services.
3. Mount the GPFS through the NFS protocol on the NFS client.

```bash
[root@hdp2 ~]# mkdir /gpfstest
[root@hdp2 ~]# mount 10.63.150.51:/gpfs1 /gpfstest
[root@hdp2 ~]# df -h
Filesystem Size Used Avail Use% Mounted on
/dev/mapper/rhel_stlrx300s6--22-root 1.1T 113G 981G 11% /
devtmpfs 126G 0 126G 0% /dev
tmpfs 126G 16K 126G 1% /dev/shm
tmpfs 126G 510M 126G 1% /run
tmpfs 126G 0 126G 0%
/sys/fs/cgroup
/dev/sdd2 197M 191M 6.6M 97% /boot
tmpfs 26G 0 26G 0% /run/user/0
10.63.150.213:/nc_volume2 95G 5.4G 90G 6% /mnt
10.63.150.51:/gpfs1 7.3T 9.1G 7.3T 1% /gpfstest
```

4. Validate the list of GPFS files in the NFS-mounted folder.

```bash
[root@hdp2 ~]# cd /gpfstest/
[root@hdp2 gpfstest]# ls
ces gpfs-ces ha testfile
[root@hdp2 gpfstest]# ls -l
total 5242882
```

5. Move the data from the GPFS-exported NFS to the NetApp NFS by using XCP.
6. Validate the GPFS files on the NFS client.
<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Type</th>
<th>Size</th>
<th>Used</th>
<th>Avail</th>
<th>Use%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mounted on</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/mapper/rhel_st1rx300s6--22-root</td>
<td>xfs</td>
<td>1.1T</td>
<td>113G</td>
<td>981G</td>
<td>11%</td>
</tr>
<tr>
<td>devtmpfs</td>
<td>devtmpfs</td>
<td>126G</td>
<td>0</td>
<td>126G</td>
<td>0%</td>
</tr>
<tr>
<td>/dev</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tmpfs</td>
<td>tmpfs</td>
<td>126G</td>
<td>16K</td>
<td>126G</td>
<td>1%</td>
</tr>
<tr>
<td>/dev/shm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tmpfs</td>
<td>tmpfs</td>
<td>126G</td>
<td>518M</td>
<td>126G</td>
<td>1%</td>
</tr>
<tr>
<td>/run</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tmpfs</td>
<td>tmpfs</td>
<td>126G</td>
<td>0</td>
<td>126G</td>
<td>0%</td>
</tr>
<tr>
<td>/sys/fs/cgroup</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/sdd2</td>
<td>xfs</td>
<td>197M</td>
<td>191M</td>
<td>6.6M</td>
<td>97%</td>
</tr>
<tr>
<td>/boot</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tmpfs</td>
<td>tmpfs</td>
<td>26G</td>
<td>0</td>
<td>26G</td>
<td>0%</td>
</tr>
<tr>
<td>/run/user/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.63.150.213:/nc_volume2</td>
<td>nfs4</td>
<td>95G</td>
<td>5.4G</td>
<td>90G</td>
<td>6%</td>
</tr>
<tr>
<td>/mnt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.63.150.51:/gpfs1</td>
<td>nfs4</td>
<td>7.3T</td>
<td>9.1G</td>
<td>7.3T</td>
<td>1%</td>
</tr>
<tr>
<td>/gpfstest</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
[root@hdp2 mnt]# df -Th
Filesystem                          Type      Size  Used  Avail Use%
Mounted on                         |           |       |       |       |      |
/dev/mapper/rhel_st1rx300s6--22-root | xfs       | 1.1T  | 113G  | 981G  | 11%  |
/devtmpfs                          | devtmpfs  | 126G  | 0     | 126G  | 0%   |
/devtmpfs                          |           |       |       |       |      |
/tmpfs                             | tmpfs     | 126G  | 16K   | 126G  | 1%   |
/dev/shm                           |           |       |       |       |      |
/tmpfs                             | tmpfs     | 126G  | 518M  | 126G  | 1%   |
/run                               |           |       |       |       |      |
/tmpfs                             | tmpfs     | 126G  | 0     | 126G  | 0%   |
/sys/fs/cgroup                     |           |       |       |       |      |
/dev/sdd2                          | xfs       | 197M  | 191M  | 6.6M  | 97%  |
/boot                              |           |       |       |       |      |
/tmpfs                             | tmpfs     | 26G   | 0     | 26G   | 0%   |
/run/user/0                        |           |       |       |       |      |
10.63.150.213:/nc_volume2          | nfs4      | 95G   | 5.4G  | 90G   | 6%   |
/mnt                              |           |       |       |       |      |
10.63.150.51:/gpfs1                | nfs4      | 7.3T  | 9.1G  | 7.3T  | 1%   |
/gpfstest                          |           |       |       |       |      |
```

```
[root@hdp2 mnt]# ls -ltrha
[1695]
```
MapR-FS to ONTAP NFS

This section provides the detailed steps needed to move MapR-FS data into ONTAP NFS by using NetApp XCP.

1. Provision three LUNs for each MapR node and give the LUNs ownership of all MapR nodes.
2. During installation, choose newly added LUNs for MapR cluster disks that are used for MapR-FS.
3. Install a MapR cluster according to the MapR 6.1 documentation.
4. Check the basic Hadoop operations using MapReduce commands such as `hadoop jar xxx`.
5. Keep customer data in MapR-FS. For example, we generated approximately a terabyte of sample data in MapR-FS by using Teragen.
6. Configure MapR-FS as NFS export.
   a. Disable the nlockmgr service on all MapR nodes.
b. Export specific folders from MapR-FS on all MapR nodes in the /opt/mapr/conf/exports file. Do not export the parent folder with different permissions when you export sub folders.
7. Refresh the MapR-FS NFS service.
root@workr-138: tmp$ maprcli nfsmgmt refreshexports
ERROR (22) - You do not have a ticket to communicate with 127.0.0.1:9998. Retry after obtaining a new ticket using maprlogin
root@workr-138: tmp$ su - mapr
[mapr@workr-138 ~]$ maprlogin password -cluster my.cluster.com
[Password for user 'mapr' at cluster 'my.cluster.com': ]
MapR credentials of user 'mapr' for cluster 'my.cluster.com' are written to '/tmp/maprticket_5000'
[mapr@workr-138 ~]$ maprcli nfsmgmt refreshexports

8. Assign a virtual IP range to a specific server or a set of servers in the MapR cluster. Then the MapR cluster assigns an IP to a specific server for NFS data access. The IPs enable high availability, which means that, if a server or network with a particular IP experiences failure, the next IP from the range of IPs can be used for NFS access.

If you would like to provide NFS access from all MapR nodes, then you can assign a set of virtual IPs to each server, and you can use the resources from each MapR node for NFS data access.
9. Check the virtual IPs assigned on each MapR node and use them for NFS data access.

```
root@workr-138: ~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
```

2: ens3f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP
group default qlen 1000
  link/ether 90:1b:0e:d1:5d:f9 brd ff:ff:ff:ff:ff:ff
  inet 10.63.150.138/24 brd 10.63.150.255 scope global noprefixroute
  ens3f0
    valid_lft forever preferred_lft forever
  inet 10.63.150.96/24 scope global secondary ens3f0:~m0
    valid_lft forever preferred_lft forever
  inet 10.63.150.97/24 scope global secondary ens3f0:~m1
    valid_lft forever preferred_lft forever
  inet6 fe80::921b:eff:fed1:5df9/64 scope link
    valid_lft forever preferred_lft forever
3: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
  link/ether 90:1b:0e:d1:af:b4 brd ff:ff:ff:ff:ff:ff
4: ens3f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
  link/ether 90:1b:0e:d1:5d:fa brd ff:ff:ff:ff:ff:ff
5: eno2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
  link/ether 90:1b:0e:d1:af:b5 brd ff:ff:ff:ff:ff:ff

[root@workr-138: ~]$ [root@workr-140 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1000
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
  inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: ens3f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP
group default qlen 1000
  link/ether 90:1b:0e:d1:5e:03 brd ff:ff:ff:ff:ff:ff
  inet 10.63.150.140/24 brd 10.63.150.255 scope global noprefixroute
  ens3f0
    valid_lft forever preferred_lft forever
  inet 10.63.150.92/24 scope global secondary ens3f0:~m0
    valid_lft forever preferred_lft forever
  inet6 fe80::921b:eff:fed1:5e03/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
3: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
  link/ether 90:1b:0e:d1:af:9a brd ff:ff:ff:ff:ff:ff
4: ens3f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
10. Mount the NFS-exported MapR-FS using the assigned virtual IP for checking the NFS operation. However, this step is not required for data transfer using NetApp XCP.

```bash
root@workr-138: tmp$ mount -v -t nfs 10.63.150.92:/maprnfs3 /tmp/testmount/
mount.nfs: timeout set for Thu Dec  5 15:31:32 2019
mount.nfs: trying text-based options
'vers=4.1,addr=10.63.150.92,clientaddr=10.63.150.138'
mount.nfs: mount(2): Protocol not supported
mount.nfs: trying text-based options
'vers=4.0,addr=10.63.150.92,clientaddr=10.63.150.138'
mount.nfs: mount(2): Protocol not supported
mount.nfs: trying text-based options 'addr=10.63.150.92'
mount.nfs: prog 100003, trying vers=3, prot=6
mount.nfs: trying 10.63.150.92 prog 100003 vers 3 prot TCP port 2049
mount.nfs: prog 100005, trying vers=3, prot=17
mount.nfs: trying 10.63.150.92 prog 100005 vers 3 prot UDP port 2049
mount.nfs: portmap query retrying: RPC: Timed out
mount.nfs: prog 100005, trying vers=3, prot=6
mount.nfs: trying 10.63.150.92 prog 100005 vers 3 prot TCP port 2049
root@workr-138: tmp$ df -h
Filesystem              Size  Used Avail Use% Mounted on
/dev/sda7                84G   48G   37G  57% /
devtmpfs                 126G     0  126G   0% /dev
tmpfs                    126G     0  126G   0% /dev/shm
tmpfs                    126G  19M  126G   1% /run
tmpfs                    126G     0  126G   0% /sys/fs/cgroup
/dev/sdd1               3.7T  201G  3.5T   6% /mnt/sdd1
/dev/sda6                946M  220M  726M  24% /boot
tmpfs                    26G     0  26G   0% /run/user/5000
gpfs1                   7.3T  9.1G  7.3T   1% /gpfs1
tmpfs                    26G     0  26G   0% /run/user/0
localhost:/mapr          100G     0  100G   0% /mapr
10.63.150.92:/maprnfs3   53T   8.4G  53T   1% /tmp/testmount
root@workr-138: tmp$
```

11. Configure NetApp XCP to transfer data from the MapR-FS NFS gateway to ONTAP NFS.
   a. Configure the catalog location for XCP.
b. Copy the license file to /opt/NetApp/xFiles/xcp/.

c. Activate XCP using the `xcp activate` command.

d. Check the source for NFS export.
e. Transfer the data using XCP from multiple MapR nodes from multiple source IPs and multiple destination IPs (ONTAP LIFs).

```bash
root@workr-138: linux$ ./xcp_yatin copy --parallel 20
10.63.150.96,10.63.150.97:/maprnfs3/tg4
10.63.150.85,10.63.150.86:/datapipline_dataset/tg4_dest
XCP 1.6-dev; (c) 2019 NetApp, Inc.; Licensed to Karthikeyan Nagalingam [NetApp Inc] until Wed Feb  5 11:07:27 2020
xcp: WARNING: No index name has been specified, creating one with
xcp: mount '10.63.150.96,10.63.150.97:/maprnfs3/tg4': WARNING: This NFS server only supports 1-second timestamp granularity. This may cause sync to fail because changes will often be undetectable.
  130 scanned, 128 giants, 3.59 GiB in (723 MiB/s), 3.60 GiB out (724 MiB/s), 5s
  130 scanned, 128 giants, 8.01 GiB in (889 MiB/s), 8.02 GiB out (890 MiB/s), 11s
  130 scanned, 128 giants, 12.6 GiB in (933 MiB/s), 12.6 GiB out (934 MiB/s), 16s
  130 scanned, 128 giants, 16.7 GiB in (830 MiB/s), 16.7 GiB out (831 MiB/s), 21s
  130 scanned, 128 giants, 21.1 GiB in (907 MiB/s), 21.1 GiB out (908 MiB/s), 26s
```
[root@workr-140 linux]# ./xcp_yatin copy --parallel 20
10.63.150.92:/maprnfs3/tg4_2
10.63.150.85,10.63.150.86:/datapipeline_dataset/tg4_2_dest
XCP 1.6-dev; (c) 2019 NetApp, Inc.; Licensed to Karthikeyan
xcp: WARNING: No index name has been specified, creating one with
xcp: mount '10.63.150.92:/maprnfs3/tg4_2': WARNING: This NFS server
only supports 1-second timestamp granularity. This may cause sync to
fail because changes will often be undetectable.
130 scanned, 128 giants, 4.39 GiB in (896 MiB/s), 4.39 GiB out (897
MiB/s), 5s
130 scanned, 128 giants, 9.94 GiB in (1.10 GiB/s), 9.96 GiB out
(1.10 GiB/s), 10s
130 scanned, 128 giants, 15.4 GiB in (1.09 GiB/s), 15.4 GiB out
(1.09 GiB/s), 15s
130 scanned, 128 giants, 20.1 GiB in (953 MiB/s), 20.1 GiB out (954
MiB/s), 20s
130 scanned, 128 giants, 24.6 GiB in (928 MiB/s), 24.7 GiB out (929
MiB/s), 25s
130 scanned, 128 giants, 29.0 GiB in (877 MiB/s), 29.0 GiB out (878
MiB/s), 31s
130 scanned, 128 giants, 33.2 GiB in (852 MiB/s), 33.2 GiB out (853
MiB/s), 36s
130 scanned, 128 giants, 37.8 GiB in (941 MiB/s), 37.8 GiB out (942
MiB/s), 41s
130 scanned, 128 giants, 42.0 GiB in (860 MiB/s), 42.0 GiB out (861
MiB/s), 46s
130 scanned, 128 giants, 46.1 GiB in (852 MiB/s), 46.2 GiB out (853
MiB/s), 51s
130 scanned, 128 giants, 50.1 GiB in (816 MiB/s), 50.2 GiB out (817
MiB/s), 56s
130 scanned, 128 giants, 54.1 GiB in (819 MiB/s), 54.2 GiB out (820
MiB/s), 1m1s
130 scanned, 128 giants, 58.5 GiB in (897 MiB/s), 58.6 GiB out (898
MiB/s), 1m6s
130 scanned, 128 giants, 62.9 GiB in (900 MiB/s), 63.0 GiB out (901
MiB/s), 1m11s
130 scanned, 128 giants, 67.2 GiB in (876 MiB/s), 67.2 GiB out (877
MiB/s), 1m16s
f. Check the load distribution on the storage controller.

```
Hadoop-AFF8080:~*> statistics show-periodic -interval 2 -iterations 0 -summary true -object nic_common -counter rx_bytes|tx_bytes -node Hadoop-AFF8080-01 -instance e3b
rx_bytes tx_bytes
-------- --------
 879MB   4.67MB
 856MB   4.46MB
 973MB   5.66MB
 986MB   5.88MB
 945MB   5.30MB
 920MB   4.92MB
 894MB   4.76MB
 902MB   4.79MB
 886MB   4.68MB
 892MB   4.78MB
 908MB   4.96MB
 905MB   4.85MB
 899MB   4.83MB
```

```
Hadoop-AFF8080:~*> statistics show-periodic -interval 2 -iterations 0 -summary true -object nic_common -counter rx_bytes|tx_bytes -node Hadoop-AFF8080-01 -instance e9b
rx_bytes tx_bytes
-------- --------
 950MB   4.93MB
 991MB   5.84MB
 959MB   5.63MB
 914MB   5.06MB
 903MB   4.81MB
 899MB   4.73MB
 892MB   4.71MB
 890MB   4.72MB
 905MB   4.86MB
 902MB   4.90MB
```

**Where to find additional information**

To learn more about the information that is described in this document, review the following documents and/or websites:

- NetApp In-Place Analytics Module Best Practices
Best practices for Confluent Kafka

TR-4912: Best practice guidelines for Confluent Kafka tiered storage with NetApp

Karthikeyan Nagalingam, Joseph Kandatilparambil, NetApp
Rankesh Kumar, Confluent

Apache Kafka is a community-distributed event-streaming platform capable of handling trillions of events a day. Initially conceived as a messaging queue, Kafka is based on an abstraction of a distributed commit log. Since it was created and open-sourced by LinkedIn in 2011, Kafka has evolved from a messages queue to a full-fledged event-streaming platform. Confluent delivers the distribution of Apache Kafka with the Confluent Platform. The Confluent Platform supplements Kafka with additional community and commercial features designed to enhance the streaming experience of both operators and developers in production at a massive scale.

This document describes the best-practice guidelines for using Confluent Tiered Storage on a NetApp’s Object storage offering by providing the following content:

- Confluent verification with NetApp Object storage – NetApp StorageGRID
- Tiered storage performance tests
- Best-practice guidelines for Confluent on NetApp storage systems

Why Confluent Tiered Storage?

Confluent has become the default real-time streaming platform for many applications, especially for big data, analytics, and streaming workloads. Tiered Storage enables users to separate compute from storage in the Confluent platform. It makes storing data more cost effective, enables you to store virtually infinite amounts of data and scale workloads up (or down) on-demand, and makes administrative tasks like data and tenant rebalancing easier. S3 compatible storage systems can take advantage of all these capabilities to democratize data with all events in one place, eliminating the need for complex data engineering. For more info on why you should use tiered storage for Kafka, check this article by Confluent.

Why NetApp StorageGRID for tiered storage?

StorageGRID is an industry-leading object storage platform by NetApp. StorageGRID is a software-defined, object-based storage solution that supports industry-standard object APIs, including the Amazon Simple Storage Service (S3) API. StorageGRID stores and manages unstructured data at scale to provide secure, durable object storage. Content is placed in the right location, at the right time, and on the right storage tier, optimizing workflows and reducing costs for globally distributed rich media.
The greatest differentiator for StorageGRID is its Information Lifecycle Management (ILM) policy engine that enables policy-driven data lifecycle management. The policy engine can use metadata to manage how data is stored across its lifetime to initially optimize for performance and automatically optimize for cost and durability as data ages.

**Enabling Confluent Tiered Storage**

The basic idea of tiered storage is to separate the tasks of data storage from data processing. With this separation, it becomes much easier for the data storage tier and the data processing tier to scale independently.

A tiered storage solution for Confluent must contend with two factors. First, it must work around or avoid common object store consistency and availability properties, such as inconsistencies in LIST operations and occasional object unavailability. Secondly, it must correctly handle the interaction between tiered storage and Kafka’s replication and fault tolerance model, including the possibility of zombie leaders continuing to tier offset ranges. NetApp Object storage provides both the consistent object availability and HA model make the tiered storage available to tier offset ranges. NetApp object storage provides consistent object availability and an HA model to make the tiered storage available to tier offset ranges.

With tiered storage, you can use high-performance platforms for low-latency reads and writes near the tail of your streaming data, and you can also use cheaper, scalable object stores like NetApp StorageGRID for high-throughput historical reads. We also have technical solution for Spark with netapp storage controller and details are here. The following figure shows how Kafka fits into a real-time analytics pipeline.

The following figure depicts how NetApp StorageGRID fits in as Confluent Kafka’s object storage tier.
Solution architecture details

This section covers the hardware and software used for Confluent verification. This information is applicable to Confluent Platform deployment with NetApp storage. The following table covers the tested solution architecture and base components.

<table>
<thead>
<tr>
<th>Solution components</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confluent Kafka version 6.2</td>
<td>• Three zookeepers&lt;br&gt;• Five broker servers&lt;br&gt;• Five tools servers&lt;br&gt;• One Grafana&lt;br&gt;• One control center</td>
</tr>
<tr>
<td>Linux (ubuntu 18.04)</td>
<td>All servers</td>
</tr>
<tr>
<td>NetApp StorageGRID for tiered storage</td>
<td>• StorageGRID software&lt;br&gt;• 1 x SG1000 (load balancer)&lt;br&gt;• 4 x SGF6024&lt;br&gt;• 4 x 24 x 800 SSDs&lt;br&gt;• S3 protocol&lt;br&gt;• 4 x 100GbE (network connectivity between broker and StorageGRID instances)</td>
</tr>
<tr>
<td>Solution components</td>
<td>Details</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------</td>
</tr>
<tr>
<td>15 Fujitsu PRIMERGY RX2540 servers</td>
<td>Each equipped with:</td>
</tr>
<tr>
<td></td>
<td>* 2 CPUs, 16 physical cores total</td>
</tr>
<tr>
<td></td>
<td>* Intel Xeon</td>
</tr>
<tr>
<td></td>
<td>* 256GB physical memory</td>
</tr>
<tr>
<td></td>
<td>* 100GbE dual port</td>
</tr>
</tbody>
</table>

**Technology overview**

This section describes the technology used in this solution.

**NetApp StorageGRID**

NetApp StorageGRID is a high-performance, cost-effective object storage platform. By using tiered storage, most of the data on Confluent Kafka, which is stored in local storage or the SAN storage of the broker, is offloaded to the remote object store. This configuration results in significant operational improvements by reducing the time and cost to rebalance, expand, or shrink clusters or replace a failed broker. Object storage plays an important role in managing data that resides on the object store tier, which is why picking the right object storage is important.

StorageGRID offers intelligent, policy-driven global data management using a distributed, node-based grid architecture. It simplifies the management of petabytes of unstructured data and billions of objects through its ubiquitous global object namespace combined with sophisticated data management features. Single-call object access extends across sites and simplifies high availability architectures while ensuring continual object access, regardless of site or infrastructure outages.

Multitenancy allows multiple unstructured cloud and enterprise data applications to be securely serviced within the same grid, increasing the ROI and use cases for NetApp StorageGRID. You can create multiple service levels with metadata-driven object lifecycle policies, optimizing durability, protection, performance, and locality across multiple geographies. Users can adjust data management policies and monitor and apply traffic limits to realign with the data landscape nondisruptively as their requirements change in ever-changing IT environments.

**Simple management with Grid Manager**

The StorageGRID Grid Manager is a browser-based graphical interface that allows you to configure, manage, and monitor your StorageGRID system across globally distributed locations in a single pane of glass.
You can perform the following tasks with the StorageGRID Grid Manager interface:

- Manage globally distributed, petabyte-scale repositories of objects such as images, video, and records.
- Monitor grid nodes and services to ensure object availability.
- Manage the placement of object data over time using information lifecycle management (ILM) rules. These rules govern what happens to an object’s data after it is ingested, how it is protected from loss, where object data is stored, and for how long.
- Monitor transactions, performance, and operations within the system.

Information Lifecycle Management policies

StorageGRID has flexible data management policies that include keeping replica copies of your objects and using EC (erasure coding) schemes like 2+1 and 4+2 (among others) to store your objects, depending on specific performance and data protection requirements. As workloads and requirements change over time, it’s common that ILM policies must change over time as well. Modifying ILM policies is a core feature, allowing StorageGRID customers to adapt to their ever-changing environment quickly and easily. Please check the ILM policy and ILM rules setup in StorageGRID.

Performance

StorageGRID scales performance by adding more storage nodes, which can be VMs, bare metal, or purpose-built appliances like the SG5712, SG5760, SG6060, or SGF6024. In our tests, we exceeded the Apache Kafka key performance requirements with a minimum-sized, three-node grid using the SGF6024 appliance. As customers scale their Kafka cluster with additional brokers, they can add more storage nodes to increase performance and capacity.

Load balancer and endpoint configuration

Admin nodes in StorageGRID provide the Grid Manager UI (user interface) and REST API endpoint to view,
configure, and manage your StorageGRID system, as well as audit logs to track system activity. To provide a highly available S3 endpoint for Confluent Kafka tiered storage, we implemented the StorageGRID load balancer, which runs as a service on admin nodes and gateway nodes. In addition, the load balancer also manages local traffic and talks to the GSLB (Global Server Load Balancing) to help with disaster recovery.

To further enhance endpoint configuration, StorageGRID provides traffic classification policies built into the admin node, lets you monitor your workload traffic, and applies various quality-of-service (QoS) limits to your workloads. Traffic classification policies are applied to endpoints on the StorageGRID Load Balancer service for gateway nodes and admin nodes. These policies can assist with traffic shaping and monitoring.

Traffic classification in StorageGRID

StorageGRID has built-in QoS functionality. Traffic classification policies can help monitor different types of S3 traffic coming from a client application. You can then create and apply policies to put limits on this traffic based on in/out bandwidth, the number of read/write concurrent requests, or the read/write request rate.

Apache Kafka

Apache Kafka is a framework implementation of a software bus using stream processing written in Java and Scala. It’s aimed to provide a unified, high-throughput, low-latency platform for handling real-time data feeds. Kafka can connect to an external system for data export and import through Kafka Connect and provides Kafka streams, a Java stream processing library. Kafka uses a binary, TCP-based protocol that is optimized for efficiency and relies on a “message set” abstraction that naturally groups messages together to reduce the overhead of the network roundtrip. This enables larger sequential disk operations, larger network packets, and contiguous memory blocks, thereby enabling Kafka to turn a bursty stream of random message writes into linear writes. The following figure depicts the basic data flow of Apache Kafka.

Kafka stores key-value messages that come from an arbitrary number of processes called producers. The data can be partitioned into different partitions within different topics. Within a partition, messages are strictly ordered by their offsets (the position of a message within a partition) and indexed and stored together with a timestamp. Other processes called consumers can read messages from partitions. For stream processing, Kafka offers the Streams API that allows writing Java applications that consume data from Kafka and write results back to Kafka. Apache Kafka also works with external stream processing systems such as Apache
Kafka runs on a cluster of one or more servers (called brokers), and the partitions of all topics are distributed across the cluster nodes. Additionally, partitions are replicated to multiple brokers. This architecture allows Kafka to deliver massive streams of messages in a fault-tolerant fashion and has allowed it to replace some of the conventional messaging systems like Java Message Service (JMS), Advanced Message Queuing Protocol (AMQP), and so on. Since the 0.11.0.0 release, Kafka offers transactional writes, which provide exactly once stream processing using the Streams API.

Kafka supports two types of topics: regular and compacted. Regular topics can be configured with a retention time or a space bound. If there are records that are older than the specified retention time or if the space bound is exceeded for a partition, Kafka is allowed to delete old data to free storage space. By default, topics are configured with a retention time of 7 days, but it’s also possible to store data indefinitely. For compacted topics, records don’t expire based on time or space bounds. Instead, Kafka treats later messages as updates to older message with the same key and guarantees never to delete the latest message per key. Users can delete messages entirely by writing a so-called tombstone message with the null value for a specific key.

There are five major APIs in Kafka:

- **Producer API.** Permits an application to publish streams of records.
- **Consumer API.** Permits an application to subscribe to topics and processes streams of records.
- **Connector API.** Executes the reusable producer and consumer APIs that can link the topics to the existing applications.
- **Streams API.** This API converts the input streams to output and produces the result.
- **Admin API.** Used to manage Kafka topics, brokers and other Kafka objects.

The consumer and producer APIs build on top of the Kafka messaging protocol and offer a reference implementation for Kafka consumer and producer clients in Java. The underlying messaging protocol is a binary protocol that developers can use to write their own consumer or producer clients in any programming language. This unlocks Kafka from the Java Virtual Machine (JVM) ecosystem. A list of available non-Java clients is maintained in the Apache Kafka wiki.

**Apache Kafka use cases**

Apache Kafka is most popular for messaging, website activity tracking, metrics, log aggregation, stream processing, event sourcing, and commit logging.

- Kafka has improved throughput, built-in partitioning, replication, and fault-tolerance, which makes it a good solution for large-scale message-processing applications.
- Kafka can rebuild a user’s activities (page views, searches) in a tracking pipeline as a set of real-time publish-subscribe feeds.
- Kafka is often used for operational monitoring data. This involves aggregating statistics from distributed applications to produce centralized feeds of operational data.
- Many people use Kafka as a replacement for a log aggregation solution. Log aggregation typically collects physical log files off of servers and puts them in a central place (for example, a file server or HDFS) for processing. Kafka abstracts files details and provides a cleaner abstraction of log or event data as a stream of messages. This allows for lower-latency processing and easier support for multiple data sources and distributed data consumption.
- Many users of Kafka process data in processing pipelines consisting of multiple stages, in which raw input data is consumed from Kafka topics and then aggregated, enriched, or otherwise transformed into new topics for further consumption or follow-up processing. For example, a processing pipeline for
recommending news articles might crawl article content from RSS feeds and publish it to an "articles" topic. Further processing might normalize or deduplicate this content and publish the cleansed article content to a new topic, and a final processing stage might attempt to recommend this content to users. Such processing pipelines create graphs of real-time data flows based on the individual topics.

- Event souring is a style of application design for which state changes are logged as a time-ordered sequence of records. Kafka’s support for very large stored log data makes it an excellent backend for an application built in this style.

- Kafka can serve as a kind of external commit-log for a distributed system. The log helps replicate data between nodes and acts as a re-syncing mechanism for failed nodes to restore their data. The log compaction feature in Kafka helps support this use case.

**Confluent**

Confluent Platform is an enterprise-ready platform that completes Kafka with advanced capabilities designed to help accelerate application development and connectivity, enable transformations through stream processing, simplify enterprise operations at scale, and meet stringent architectural requirements. Built by the original creators of Apache Kafka, Confluent expands the benefits of Kafka with enterprise-grade features while removing the burden of Kafka management or monitoring. Today, over 80% of the Fortune 100 are powered by data streaming technology – and most of those use Confluent.

**Why Confluent?**

By integrating historical and real-time data into a single, central source of truth, Confluent makes it easy to build an entirely new category of modern, event-driven applications, gain a universal data pipeline, and unlock powerful new use cases with full scalability, performance, and reliability.

**What is Confluent used for?**

Confluent Platform lets you focus on how to derive business value from your data rather than worrying about the underlying mechanics, such as how data is being transported or integrated between disparate systems. Specifically, Confluent Platform simplifies connecting data sources to Kafka, building streaming applications, as well as securing, monitoring, and managing your Kafka infrastructure. Today, Confluent Platform is used for a wide array of use cases across numerous industries, from financial services, omnichannel retail, and autonomous cars, to fraud detection, microservices, and IoT.

The following figure shows Confluent Kafka Platform components.
Overview of Confluent's event streaming technology

At the core of Confluent Platform is Apache Kafka, the most popular open-source distributed streaming platform. The key capabilities of Kafka are as follows:

- Publish and subscribe to streams of records.
- Store streams of records in a fault tolerant way.
- Process streams of records.

Out of the box, Confluent Platform also includes Schema Registry, REST Proxy, a total of 100+ prebuilt Kafka connectors, and ksqlDB.

Overview of Confluent platform’s enterprise features

- **Confluent Control Center.** A GUI-based system for managing and monitoring Kafka. It allows you to easily manage Kafka Connect and to create, edit, and manage connections to other systems.

- **Confluent for Kubernetes.** Confluent for Kubernetes is a Kubernetes operator. Kubernetes operators extend the orchestration capabilities of Kubernetes by providing the unique features and requirements for a specific platform application. For Confluent Platform, this includes greatly simplifying the deployment process of Kafka on Kubernetes and automating typical infrastructure lifecycle tasks.

- **Confluent connectors to Kafka.** Connectors use the Kafka Connect API to connect Kafka to other systems such as databases, key-value stores, search indexes, and file systems. Confluent Hub has downloadable connectors for the most popular data sources and sinks, including fully tested and supported versions of these connectors with Confluent Platform. More details can be found here.

- **Self-balancing clusters.** Provides automated load balancing, failure detection and self-healing. It provides support for adding or decommissioning brokers as needed, with no manual tuning.

- **Confluent cluster linking.** Directly connects clusters together and mirrors topics from one cluster to another over a link bridge. Cluster linking simplifies setup of multi-datacenter, multi-cluster, and hybrid
cloud deployments.

- **Confluent auto data balancer.** Monitors your cluster for the number of brokers, the size of partitions, number of partitions, and the number of leaders within the cluster. It allows you to shift data to create an even workload across your cluster, while throttling rebalance traffic to minimize the effect on production workloads while rebalancing.

- **Confluent replicator.** Makes it easier than ever to maintain multiple Kafka clusters in multiple data centers.

- **Tiered storage.** Provides options for storing large volumes of Kafka data using your favorite cloud provider, thereby reducing operational burden and cost. With tiered storage, you can keep data on cost-effective object storage and scale brokers only when you need more compute resources.

- **Confluent JMS client.** Confluent Platform includes a JMS-compatible client for Kafka. This Kafka client implements the JMS 1.1 standard API, using Kafka brokers as the backend. This is useful if you have legacy applications using JMS and you would like to replace the existing JMS message broker with Kafka.

- **Confluent MQTT proxy.** Provides a way to publish data directly to Kafka from MQTT devices and gateways without the need for a MQTT broker in the middle.

- **Confluent security plugins.** Confluent security plugins are used to add security capabilities to various Confluent Platform tools and products. Currently, there is a plugin available for the Confluent REST proxy that helps to authenticate the incoming requests and propagate the authenticated principal to requests to Kafka. This enables Confluent REST proxy clients to utilize the multitenant security features of the Kafka broker.

### Confluent verification

We performed verification with Confluent Platform 6.2 Tiered Storage in NetApp StorageGRID. The NetApp and Confluent teams worked on this verification together and ran the test cases required for verification.

### Confluent Platform setup

We used the following setup for verification.

For verification, we used three zookeepers, five brokers, five test-script executing servers, named tools servers with 256GB RAM, and 16 CPUs. For NetApp storage, we used StorageGRID with an SG1000 load balancer with four SGF6024s. The storage and brokers were connected via 100GbE connections.

The following figure shows the network topology of configuration used for Confluent verification.
The tools servers act as application clients that send requests to Confluent nodes.

**Confluent tiered storage configuration**

The tiered storage configuration requires the following parameters in Kafka:

```yaml
Confluent.tier.archiver.num.threads=16
confluent.tier.fetcher.num.threads=32
confluent.tier.enable=true
confluent.tier.feature=true
confluent.tier.backend=S3
confluent.tier.s3.bucket=kafkasgdbucket1-2
confluent.tier.s3.region=us-west-2
confluent.tier.s3.cred.file.path=/data/kafka/.ssh/credentials
confluent.tier.s3.aws.endpoint.override=http://kafkasgd.rtpppe.netapp.com:10444/
confluent.tier.s3.force.path.style.access=true
```

For verification, we used StorageGRID with the HTTP protocol, but HTTPS also works. The access key and secret key are stored in the file name provided in the `confluent.tier.s3.cred.file.path` parameter.

**NetApp object storage - StorageGRID**

We configured single-site configuration in StorageGRID for verification.
Verification tests

We completed the following five test cases for the verification. These tests are executed on the Trogdor framework. The first two were functionality tests and the remaining three were performance tests.
Object store correctness test

This test determines whether all basic operations (for example, get/put/delete) on the object store API work well according to the needs of tiered storage. It is a basic test that every object store service should expect to pass ahead of the following tests. It is an assertive test that either passes or fails.

Tiering functionality correctness test

This test determines if end-to-end tiered storage functionality works well with an assertive test that either passes or fails. The test creates a test topic that by default is configured with tiering enabled and highly a reduced hotset size. It produces an event stream to the newly created test topic, it waits for the brokers to archive the segments to the object store, and it then consumes the event stream and validates that the consumed stream matches the produced stream. The number of messages produced to the event stream is configurable, which lets the user generate a sufficiently large workload according to the needs of testing. The reduced hotset size ensures that the consumer fetches outside the active segment are served only from the object store; this helps test the correctness of the object store for reads. We have performed this test with and without an object-store fault injection. We simulated node failure by stopping the service manager service in one of the nodes in StorageGRID and validating that the end-to-end functionality works with object storage.

Tier fetch benchmark

This test validated the read performance of the tiered object storage and checked the range fetch read requests under heavy load from segments generated by the benchmark. In this benchmark, Confluent developed custom clients to serve the tier fetch requests.

Produce-consume workload benchmark

This test indirectly generated write workload on the object store through the archival of segments. The read workload (segments read) was generated from object storage when consumer groups fetched the segments. This workload was generated by the test script. This test checked the performance of read and write on the object storage in parallel threads. We tested with and without object store fault injection as we did for the tiering functionality correctness test.

Retention workload benchmark

This test checked the deletion performance of an object store under a heavy topic-retention workload. The retention workload was generated using a test script that produces many messages in parallel to a test topic. The test topic was configuring with an aggressive size-based and time-based retention setting that caused the event stream to be continuously purged from the object store. The segments were then archived. This led to a large number of deletions in the object storage by the broker and collection of the performance of the object-store delete operations.

Performance tests with scalability

We performed the tiered storage testing with three to four nodes for producer and consumer workloads with the NetApp StorageGRID setup. According to our tests, the time to completion and the performance results were directly proportional to the number of StorageGRID nodes. The StorageGRID setup required a minimum of three nodes.

• The time to complete the produce and consumer operation decreased linearly when the number of storage nodes increased.
The performance for the s3 retrieve operation increased linearly based on number of StorageGRID nodes. StorageGRID supports up to 200 StorageGRID nodes.
Confluent s3 connector

The Amazon S3 Sink connector exports data from Apache Kafka topics to S3 objects in either the Avro, JSON, or Bytes formats. The Amazon S3 sink connector periodically polls data from Kafka and in turn uploads it to S3. A partitioner is used to split the data of every Kafka partition into chunks. Each chunk of data is represented as an S3 object. The key name encodes the topic, the Kafka partition, and the start offset of this data chunk.

In this setup, we show you how to read and write topics in object storage from Kafka directly using the Kafka s3 sink connector. For this test, we used a stand-alone Confluent cluster, but this setup is applicable to a distributed cluster.

1. Download Confluent Kafka from the Confluent website.
2. Unpack the package to a folder on your server.
3. Export two variables.

   Export  CONFLUENT_HOME=/data/confluent/confluent-6.2.0
   export  PATH=$PATH:/data/confluent/confluent-6.2.0/bin

4. For a stand-alone Confluent Kafka setup, the cluster creates a temporary root folder in /tmp. It also creates Zookeeper, Kafka, a schema registry, connect, a ksql-server, and control-center folders and copies their respective configuration files from $CONFLUENT_HOME. See the following example:

   root@stlr2540m1-108:~# ls -ltr /tmp/confluent.406980/
   total 28
   drwxr-xr-x 4 root root 4096 Oct 29 19:01 zookeeper
   drwxr-xr-x 4 root root 4096 Oct 29 19:37 kafka
   drwxr-xr-x 4 root root 4096 Oct 29 19:40 schema-registry
   drwxr-xr-x 4 root root 4096 Oct 29 19:45 kafka-rest
   drwxr-xr-x 4 root root 4096 Oct 29 19:47 connect
   drwxr-xr-x 4 root root 4096 Oct 29 19:48 ksql-server
   drwxr-xr-x 4 root root 4096 Oct 29 19:53 control-center
   root@stlr2540m1-108:~#

5. Configure Zookeeper. You don’t need to change anything if you use the default parameters.
In the above configuration, we updated the `server.xxx` property. By default, you need three Zookeepers for the Kafka leader selection.

6. We created a myid file in `/tmp/confluent.406980/zookeeper/data` with a unique ID:

```
root@stlrx2540m1-108:~# cat /tmp/confluent.406980/zookeeper/data/myid
179
```

We used the last number of IP addresses for the myid file. We used default values for the Kafka, connect, control-center, Kafka, Kafka-rest, ksql-server, and schema-registry configurations.

7. Start the Kafka services.

```
root@stlrx2540m1-108:/data/confluent/confluent-6.2.0/bin# confluent local services start
The local commands are intended for a single-node development environment only,
NOT for production usage.

Using CONFLUENT_CURRENT: /tmp/confluent.406980
ZooKeeper is [UP]
Kafka is [UP]
Schema Registry is [UP]
Kafka REST is [UP]
Connect is [UP]
ksqlDB Server is [UP]
Control Center is [UP]
```

There is a log folder for each configuration, which helps troubleshoot issues. In some instances, services take more time to start. Make sure all services are up and running.
8. **Install Kafka connect using** `confluent-hub`.

```
root@stlrx2540m1-108:/data/confluent/confluent-6.2.0/bin# ./confluent-hub install confluentinc/kafka-connect-s3:latest
The component can be installed in any of the following Confluent Platform installations:
  1. /data/confluent/confluent-6.2.0 (based on $CONFLUENT_HOME)
  2. /data/confluent/confluent-6.2.0 (where this tool is installed)
Choose one of these to continue the installation (1-2): 1
Do you want to install this into /data/confluent/confluent-6.2.0/share/confluent-hub-components? (yN) y
Component's license:
Confluent Community License
http://www.confluent.io/confluent-community-license
I agree to the software license agreement (yN) y
Downloading component Kafka Connect S3 10.0.3, provided by Confluent, Inc. from Confluent Hub and installing into /data/confluent/confluent-6.2.0/share/confluent-hub-components
Do you want to uninstall existing version 10.0.3? (yN) y
Detected Worker's configs:
  1. Standard: /data/confluent/confluent-6.2.0/etc/kafka/connect-distributed.properties
  2. Standard: /data/confluent/confluent-6.2.0/etc/kafka/connect-standalone.properties
  3. Standard: /data/confluent/confluent-6.2.0/etc/schema-registry/connect-avro-distributed.properties
  4. Standard: /data/confluent/confluent-6.2.0/etc/schema-registry/connect-avro-standalone.properties
  5. Based on CONFLUENT_CURRENT:
     /tmp/confluent.406980/connect/connect.properties
  6. Used by Connect process with PID 15904:
     /tmp/confluent.406980/connect/connect.properties
Do you want to update all detected configs? (yN) y
Adding installation directory to plugin path in the following files:
  /data/confluent/confluent-6.2.0/etc/kafka/connect-distributed.properties
  /data/confluent/confluent-6.2.0/etc/kafka/connect-standalone.properties
  /data/confluent/confluent-6.2.0/etc/schema-registry/connect-avro-distributed.properties
  /data/confluent/confluent-6.2.0/etc/schema-registry/connect-avro-standalone.properties
  /tmp/confluent.406980/connect/connect.properties
  /tmp/confluent.406980/connect/connect.properties
```
You can also install a specific version by using `confluent-hub install confluentinc/kafka-connect-s3:10.0.3`.

9. By default, `confluentinc-kafka-connect-s3` is installed in `/data/confluent/confluent-6.2.0/share/confluent-hub-components/confluentinc-kafka-connect-s3`.

10. Update the plug-in path with the new `confluentinc-kafka-connect-s3`.

```
root@stlrx2540m1-108:/data/confluent/confluent-6.2.0/etc/kafka/connect-distributed.properties | grep plugin.path
# plugin.path=/usr/local/share/java,/usr/local/share/kafka/plugins,/opt/connectors,
plugin.path=/usr/share/java,/data/zookeeper/confluent/confluent-6.2.0/share/confluent-hub-components,/data/confluent/confluent-6.2.0/share/confluent-hub-components,/data/confluent/confluent-6.2.0/share/confluent-hub-components/confluentinc-kafka-connect-s3
```

11. Stop the Confluent services and restart them.

```
confluent local services stop
confluent local services start
```

The local commands are intended for a single-node development environment only, NOT for production usage.

```
Using CONFLUENT_CURRENT: /tmp/confluent.406980
Connect is [UP]
Control Center is [UP]
Kafka is [UP]
Kafka REST is [UP]
ksqldb Server is [UP]
Schema Registry is [UP]
ZooKeeper is [UP]
```

12. Configure the access ID and secret key in the `/root/.aws/credentials` file.
13. Verify that the bucket is reachable.

```bash
root@stlrx2540m4-01:~# aws s3 -endpoint-url http://kafkasgd.rtpppe.netapp.com:10444 ls kafkasgdbucket1-2
2021-10-29 21:04:18       1388 1
2021-10-29 21:04:20       1388 2
2021-10-29 21:04:22       1388 3
root@stlrx2540m4-01:~#
```

14. Configure the s3-sink properties file for s3 and bucket configuration.

```bash
root@stlrx2540m1-108:~# cat /data/confluent/confluent-6.2.0/share/confluent-hub-components/confluentinc-kafka-connect-s3/etc/quickstart-s3.properties | grep -v ^#
name=s3-sink
carrier.class=io.confluent.connect.s3.S3SinkConnector
tasks.max=1
topics=s3_testtopic
s3.region=us-west-2
s3.bucket.name=kafkasgdbucket1-2
store.url=http://kafkasgd.rtpppe.netapp.com:10444/
s3.part.size=5242880
flush.size=3
storage.class=io.confluent.connect.s3.storage.S3Storage
format.class=io.confluent.connect.s3.format.avro.AvroFormat
partitioner.class=io.confluent.connect.storage.partitioner.DefaultPartitioner
schema.compatibility=NONE
root@stlrx2540m1-108:~#
```

15. Import a few records to the s3 bucket.
16. Load the s3-sink connector.
The local commands are intended for a single-node development environment only, NOT for production usage.

https://docs.confluent.io/current/cli/index.html

```json
{
  "name": "s3-sink",
  "config": {
    "connector.class": "io.confluent.connect.s3.S3SinkConnector",
    "flush.size": "3",
    "format.class": "io.confluent.connect.s3.format.avro.AvroFormat",
    "partitioner.class": "io.confluent.connect.storage.partitioner.DefaultPartitioner",
    "s3.bucket.name": "kafkasgdbucket1-2",
    "s3.part.size": "5242880",
    "s3.region": "us-west-2",
    "schema.compatibility": "NONE",
    "storage.class": "io.confluent.connect.s3.storage.S3Storage",
    "store.url": "http://kafkasgd.rtpppe.netapp.com:10444/",
    "tasks.max": "1",
    "topics": "s3_testtopic",
    "name": "s3-sink"
  },
  "tasks": [],
  "type": "sink"
}
```

17. Check the s3-sink status.
18. Check the log to make sure that s3-sink is ready to accept topics.

```bash
root@stlrx2540m1-108:~# confluent local services connect log
```

19. Check the topics in Kafka.

```bash
kafka-topics --list --bootstrap-server localhost:9092
...
connect-configs
connect-offsets
connect-statuses
default_ksql_processing_log
s3_testtopic
s3_topic
s3_topic_new
root@stlrx2540m1-108:~#
```

20. Check the objects in the s3 bucket.
To verify the contents, copy each file from S3 to your local filesystem by running the following command:

```bash
aws s3 --endpoint-url http://kafkasgd.rtpppe.netapp.com:10444 cp s3://kafkasgdbucket1-2/topics/s3_testtopic/partition=0/s3_testtopic+0+0000000000.avro tes.avro
```

To print the records, use avro-tools-1.11.0.1.jar (available in the Apache Archives).

```bash
java -jar /usr/src/avro-tools-1.11.0.1.jar tojson tes.avro
```

Confluent Self-balancing Clusters

If you have managed a Kafka cluster before, you are likely familiar with the challenges
that come with manually reassigning partitions to different brokers to make sure that the workload is balanced across the cluster. For organizations with large Kafka deployments, reshuffling large amounts of data can be daunting, tedious, and risky, especially if mission-critical applications are built on top of the cluster. However, even for the smallest Kafka use cases, the process is time consuming and prone to human error.

In our lab, we tested the Confluent self-balancing clusters feature, which automates rebalancing based on cluster topology changes or uneven load. The Confluent rebalance test helps to measure the time to add a new broker when node failure or the scaling node requires rebalancing data across brokers. In classic Kafka configurations, the amount of data to be rebalanced grows as the cluster grows, but, in tiered storage, rebalancing is restricted to a small amount of data. Based on our validation, rebalancing in tiered storage takes seconds or minutes in a classic Kafka architecture and grows linearly as the cluster grows.

In self-balancing clusters, partition rebalances are fully automated to optimize Kafka’s throughput, accelerate broker scaling, and reduce the operational burden of running a large cluster. At steady-state, self-balancing clusters monitor the skew of data across the brokers and continuously reassigns partitions to optimize cluster performance. When scaling the platform up or down, self-balancing clusters automatically recognize the presence of new brokers or the removal of old brokers and trigger a subsequent partition reassignment. This enables you to easily add and decommission brokers, making your Kafka clusters fundamentally more elastic. These benefits come without any need for manual intervention, complex math, or the risk of human error that partition reassignments typically entail. As a result, data rebalances are completed in far less time, and you are free to focus on higher-value event-streaming projects rather than needing to constantly supervise your clusters.

**Best practice guidelines**

This section presents lessons learned from this certification.

- Based on our validation, S3 object storage is best for Confluent to keep data.
- We can use high-throughput SAN (specifically FC) to keep the broker hot data or local disk, because, in the Confluent tiered storage configuration, the size of the data held in the brokers data directory is based on the segment size and retention time when the data is moved to object storage.
- Object stores provide better performance when segment.bytes is higher; we tested 512MB.
- In Kafka, the length of the key or value (in bytes) for each record produced to the topic is controlled by the `length.key.value` parameter. For StorageGRID, S3 object ingest and retrieve performance increased to higher values. For example, 512 bytes provided a 5.8GBps retrieve, 1024 bytes provided a 7.5GBps s3 retrieve, and 2048 bytes provided close to 10GBps.

The following figure presents the S3 object ingest and retrieve based on `length.key.value`.  

1730
• **Kafka tuning.** To improve the performance of tiered storage, you can increase TierFetcherNumThreads and TierArchiverNumThreads. As a general guideline, you want to increase TierFetcherNumThreads to match the number of physical CPU cores and increase TierArchiverNumThreads to half the number of CPU cores. For example, in server properties, if you have a machine with eight physical cores, set confluent.tier.fetcher.num.threads = 8 and confluent.tier.archiver.num.threads = 4.

• **Time interval for topic deletes.** When a topic is deleted, deletion of the log segment files in object storage does not immediately begin. Rather, there is a time interval with a default value of 3 hours before deletion of those files takes place. You can modify the configuration, confluent.tier.topic.delete.check.interval.ms, to change the value of this interval. If you delete a topic or cluster, you can also manually delete the objects in the respective bucket.

• **ACLs on tiered storage internal topics.** A recommended best practice for on-premises deployments is to enable an ACL authorizer on the internal topics used for tiered storage. Set ACL rules to limit access on this data to the broker user only. This secures the internal topics and prevents unauthorized access to tiered storage data and metadata.

```bash
kafka-acls --bootstrap-server localhost:9092 --command-config adminclient-configs.conf \
--add --allow-principal User:<kafka> --operation All --topic "_confluent-tier-state"
```

Replace the user `<kafka>` with the actual broker principal in your deployment.

For example, the command `confluent-tier-state` sets ACLs on the internal topic for tiered storage. Currently, there is only a single internal topic related to tiered storage. The example creates an ACL that provides the principal Kafka permission for all operations on the internal topic.

**Sizing**

Kafka sizing can be performed with four configuration modes: simple, granular, reverse, and partitions.
Simple

The simple mode is appropriate for the first-time Apache Kafka users or early state use cases. For this mode, you provide requirements such as throughput MBps, read fanout, retention, and the resource utilization percentage (60% is default). You also enter the environment, such as on-premises (bare-metal, VMware, Kubernetes, or OpenStack) or cloud. Based on this information, the sizing of a Kafka cluster provides the number of servers required for the broker, the zookeeper, Apache Kafka connect workers, the schema registry, a REST Proxy, ksqlDB, and the Confluent control center.

For tiered storage, consider the granular configuration mode for sizing a Kafka cluster. Granular mode is appropriate for experienced Apache Kafka users or well-defined use cases. This section describes sizing for producers, stream processors, and consumers.

Producers

To describe the producers for Apache Kafka (for example a native client, REST proxy, or Kafka connector), provide the following information:

• **Name.** Spark.

• **Producer type.** Application or service, proxy (REST, MQTT, other), and existing database (RDBMS, NOSQL, other). You can also select “I don’t know.”

• **Average throughput.** In events per second (1,000,000 for example).

• **Peak throughput.** In events per second (4,000,000 for example).

• **Average message size.** In bytes, uncompressed (max 1MB; 1000 for example).

• **Message format.** Options include Avro, JSON, protocol buffers, binary, text, “I don’t know,” and other.

• **Replication factor.** Options are 1, 2, 3 (Confluent recommendation), 4, 5, or 6.

• **Retention time.** One day (for example). How long do you want your data to be stored in Apache Kafka? Enter -1 with any unit for an infinite time. The calculator assumes a retention time of 10 years for infinite retention.

• Select the check box for "Enable Tiered Storage to Decrease Broker Count and Allow for Infinite Storage?"

• When tiered storage is enabled, the retention fields control the hot set of data that is stored locally on the broker. The archival retention fields control how long data is stored in archival object storage.

• **Archival Storage Retention.** One year (for example). How long do you want your data to be stored in archival storage? Enter -1 with any unit for an infinite duration. The calculator assumes a retention of 10 years for infinite retention.

• **Growth Multiplier.** 1 (for example). If the value of this parameter is based on current throughput, set it to 1. To size based on additional growth, set this parameter to a growth multiplier.

• **Number of producer instances.** 10 (for example). How many producer instances will be running? This input is required to incorporate the CPU load into the sizing calculation. A blank value indicates that CPU load is not incorporated into the calculation.

Based on this example input, sizing has the following effect on producers:

• Average throughput in uncompressed bytes: 1GBps. Peak throughput in uncompressed bytes: 4GBps. Average throughput in compressed bytes: 400MBps. Peak throughput in compressed bytes: 1.6GBps. This is based on a default 60% compression rate (you can change this value).

  • Total on-broker hotset storage required: 31,104TB, including replication, compressed. Total off-broker archival storage required: 378,432TB, compressed. Use [https://fusion.netapp.com](https://fusion.netapp.com) for StorageGRID sizing.
Stream Processors must describe their applications or services that consume data from Apache Kafka and produce back into Apache Kafka. In most cases these are built in ksqlDB or Kafka Streams.

- **Name.** Spark streamer.
- **Processing time.** How long does this processor take to process a single message?
  - 1 ms (simple, stateless transformation) [example], 10ms (stateful in-memory operation).
  - 100ms (stateful network or disk operation), 1000ms (3rd party REST call).
  - I have benchmarked this parameter and know exactly how long it takes.
- **Output Retention.** 1 day (example). A stream processor produces its output back to Apache Kafka. How long do you want this output data to be stored in Apache Kafka? Enter -1 with any unit for an infinite duration.
  - Select the check box "Enable Tiered Storage to Decrease Broker Count and Allow for Infinite Storage?"
- **Archival Storage Retention.** 1 year (for example). How long do you want your data to be stored in archival storage? Enter -1 with any unit for an infinite duration. The calculator assumes a retention of 10 years for infinite retention.
- **Output Passthrough Percentage.** 100 (for example). A stream processor produces its output back to Apache Kafka. What percentage of inbound throughput will be outputted back into Apache Kafka? For example, if inbound throughput is 20MBps and this value is 10, the output throughput will be 2MBps.
  - Some other benchmarked process of known duration.
- **From which applications does this read from?** Select “Spark,” the name used in producer type-based sizing.

Based on the above input, you can expect the following effects of sizing on stream processor instances and topic partition estimates:

- This stream processor application requires the following number of instances. The incoming topics likely require this many partitions as well. Contact Confluent to confirm this parameter.
  - 1,000 for average throughput with no growth multiplier
  - 4,000 for peak throughput with no growth multiplier
  - 1,000 for average throughput with a growth multiplier
  - 4,000 for peak throughput with a growth multiplier

**Consumers**

Describe your applications or services that consume data from Apache Kafka and do not produce back into Apache Kafka; for example, a native client or Kafka Connector.

- **Name.** Spark consumer.
- **Processing time.** How long does this consumer take to process a single message?
  - 1ms (for example, a simple and stateless task like logging)
  - 10ms (fast writes to a datastore)
  - 100ms (slow writes to a datastore)
  - 1000ms (third party REST call)
  - Some other benchmarked process of known duration.
- **Consumer type.** Application, proxy, or sink to an existing datastore (RDBMS, NoSQL, other).
  - From which applications does this read from? Connect this parameter with producer and stream sizing determined previously.
Based on the above input, you must determine the sizing for consumer instances and topic partition estimates. A consumer application requires the following number of instances:

- 2,000 for average throughput, no growth multiplier
- 8,000 for peak throughput, no growth multiplier
- 2,000 for average throughput, including growth multiplier
- 8,000 for peak throughput, including growth multiplier

The incoming topics likely need this number of partitions as well. Contact Confluent to confirm.

In addition to the requirements for producers, stream processors, and consumers, you must provide the following additional requirements:

- **Rebuild time.** For example, 4 hours. If an Apache Kafka broker host fails, its data is lost, and a new host is provisioned to replace the failed host, how fast must this new host rebuild itself? Leave this parameter blank if the value is unknown.

- **Resource utilization target (percentage).** For example, 60. How utilized do you want your hosts to be during average throughput? Confluent recommends 60% utilization unless you are using Confluent self-balancing clusters, in which case utilization can be higher.

**Describe your environment**

- **What environment will your cluster be running in?** Amazon Web Services, Microsoft Azure, Google cloud platform, bare-metal on premises, VMware on premises, OpenStack on premises, or Kubernetes on premises?

- **Host details.** Number of cores: 48 (for example), network card type (10GbE, 40GbE, 16GbE, 1GbE, or another type).

- **Storage volumes.** Host: 12 (for example). How many hard drives or SSDs are supported per host? Confluent recommends 12 hard drives per host.

- **Storage capacity/volume (in GB).** 1000 (for example). How much storage can a single volume store in gigabytes? Confluent recommends 1TB disks.

- **Storage configuration.** How are storage volumes configured? Confluent recommends RAID10 to take advantage of all Confluent features. JBOD, SAN, RAID 1, RAID 0, RAID 5, and other types are also supported.

- **Single volume throughput (MBps).** 125 (for example). How fast can a single storage volume read or write in megabytes per second? Confluent recommends standard hard drives, which typically have 125MBps throughput.

- **Memory capacity (GB).** 64 (for example).

After you have determined your environmental variables, select Size my Cluster. Based on the example parameters indicated above, we determined the following sizing for Confluent Kafka:

- **Apache Kafka.** Broker count: 22. Your cluster is storage-bound. Consider enabling tiered storage to decrease your host count and allow for infinite storage.

- **Apache ZooKeeper.** Count: 5; Apache Kafka Connect Workers: Count: 2; Schema Registry: Count: 2; REST Proxy: Count: 2; ksqlDB: Count: 2; Confluent Control Center: Count: 1.

Use reverse mode for platform teams without a use case in mind. Use partitions mode to calculate how many partitions a single topic requires. See [https://eventsizer.io](https://eventsizer.io) for sizing based on the reverse and partitions modes.
Conclusion

This document provides best practice guidelines for using Confluent Tiered Storage with NetApp storage, including verification tests, tiered storage performance results, tuning, Confluent S3 connectors, and the self-balancing feature. Considering ILM policies, Confluent performance with multiple performance tests for verification, and industry-standard S3 APIs, NetApp StorageGRID object storage is an optimal choice for Confluent tiered storage.

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

- What is Apache Kafka
  https://www.confluent.io/what-is-apache-kafka/
- NetApp Product Documentation
  https://www.netapp.com/support-and-training/documentation/
- S3-sink parameter details
  https://docs.confluent.io/kafka-connect-s3-sink/current/configuration_options.html#s3-configuration-options
- Apache Kafka
  https://en.wikipedia.org/wiki/Apache_Kafka
- Infinite Storage in Confluent Platform
  https://www.confluent.io/blog/infinite-kafka-storage-in-confluent-platform/
- Confluent Tiered Storage - Best practices and sizing
  https://docs.confluent.io/platform/current/kafka/tiered-storage.html#best-practices-and-recommendations
- Amazon S3 sink connector for Confluent Platform
  https://docs.confluent.io/kafka-connect-s3-sink/current/overview.html
- Kafka sizing
  https://eventsizer.io
- StorageGRID sizing
  https://fusion.netapp.com/
- Kafka use cases
  https://kafka.apache.org/uses
NetApp hybrid cloud data solutions - Spark and Hadoop based on customer use cases

TR-4657: NetApp hybrid cloud data solutions - Spark and Hadoop based on customer use cases

Karthikeyan Nagalingam and Sathish Thyagarajan, NetApp

This document describes hybrid cloud data solutions using NetApp AFF and FAS storage systems, NetApp Cloud Volumes ONTAP, NetApp connected storage, and NetApp FlexClone technology for Spark and Hadoop. These solution architectures allow customers to choose an appropriate data protection solution for their environment. NetApp designed these solutions based on interaction with customers and their business use-cases. This document provides the following detailed information:

- Why we need data protection for Spark and Hadoop environments and customer challenges.
- The data fabric powered by NetApp vision and its building blocks and services.
- How these building blocks can be used to architect flexible data protection workflows.
- The pros and cons of several architectures based on real-world customer use cases. Each use case provides the following components:
  - Customer scenarios
  - Requirements and challenges
  - Solutions
  - Summary of the solutions

Why Hadoop data protection?

In a Hadoop and Spark environment, the following concerns must be addressed:

- **Software or human failures.** Human error in software updates while carrying out Hadoop data operations can lead to faulty behavior that can cause unexpected results from the job. In such case, we need to protect the data to avoid failures or unreasonable outcomes. For example, as the result of a poorly executed software update to a traffic signal analysis application, a new feature that fails to properly analyze traffic signal data in the form of plain text. The software still analyzes JSON and other non-text file formats, resulting in the real-time traffic control analytics system producing prediction results that are missing data points. This situation can cause faulty outputs that might lead to accidents at the traffic signals. Data protection can address this issue by providing the capability to quickly roll back to the previous working application version.

- **Size and scale.** The size of the analytics data grows day by day due to the ever-increasing numbers of data sources and volume. Social media, mobile apps, data analytics, and cloud computing platforms are the main sources of data in the current big data market, which is increasing very rapidly, and therefore the
data needs to be protected to ensure accurate data operations.

- **Hadoop’s native data protection.** Hadoop has a native command to protect the data, but this command does not provide consistency of data during backup. It only supports directory-level backup. The snapshots created by Hadoop are read-only and cannot be used to reuse the backup data directly.

**Data protection challenges for Hadoop and Spark customers**

A common challenge for Hadoop and Spark customers is to reduce the backup time and increase backup reliability without negatively affecting performance at the production cluster during data protection.

Customers also need to minimize recovery point objective (RPO) and recovery time objective (RTO) downtime and control their on-premises and cloud-based disaster recovery sites for optimal business continuity. This control typically comes from having enterprise-level management tools.

The Hadoop and Spark environments are complicated because not only is the data volume huge and growing, but the rate this data arrives is increasing. This scenario makes it difficult to rapidly create efficient, up-to-date DevTest and QA environments from the source data. NetApp recognizes these challenges and offers the solutions presented in this paper.

**Data fabric powered by NetApp for big data architecture**

The data fabric powered by NetApp simplifies and integrates data management across cloud and on-premises environments to accelerate digital transformation.

The data fabric powered by NetApp delivers consistent and integrated data management services and applications (building blocks) for data visibility and insights, data access and control, and data protection and security, as shown in the figure below.
Proven data fabric customer use cases

The data fabric powered by NetApp provides the following nine proven use cases for customers:

- Accelerate analytics workloads
- Accelerate DevOps transformation
- Build cloud hosting infrastructure
- Integrate cloud data services
- Protect and secure data
- Optimize unstructured data
- Gain data center efficiencies
- Deliver data insights and control
- Simplify and automate

This document covers two of the nine use cases (along with their solutions):
• Accelerate analytics workloads
• Protect and secure data

NetApp NFS direct access

The NetApp NFS allows customers to run big data analytics jobs on their existing or new NFSv3 or NFSv4 data without moving or copying the data. It prevents multiple copies of data and eliminates the need to sync the data with a source. For example, in the financial sector, the movement of data from one place to another place must meet legal obligations, which is not an easy task. In this scenario, the NetApp NFS direct access analyzes the financial data from its original location. Another key benefit is that using the NetApp NFS direct access simplifies protecting Hadoop data by using native Hadoop commands and enables data protection workflows leveraging NetApp's rich data management portfolio.

The NetApp NFS direct access provides two kinds of deployment options for Hadoop/Spark clusters:

• By default, the Hadoop/Spark clusters use Hadoop Distributed File System (HDFS) for data storage and the default file system. The NetApp NFS direct access can replace the default HDFS with NFS storage as the default file system, enabling direct analytics operations on NFS data.

• In another deployment option, the NetApp NFS direct access supports configuring NFS as additional storage along with HDFS in a single Hadoop/Spark cluster. In this case, the customer can share data through NFS exports and access it from the same cluster along with HDFS data.

The key benefits of using the NetApp NFS direct access include:

• Analyzes the data from its current location, which prevents the time- and performance-consuming task of moving analytics data to a Hadoop infrastructure such as HDFS.

• Reduces the number of replicas from three to one.

• Enables users to decouple the compute and storage to scale them independently.

• Provides enterprise data protection by leveraging the rich data management capabilities of ONTAP.
• Is certified with the Hortonworks data platform.
• Enables hybrid data analytics deployments.
• Reduces the backup time by leveraging dynamic multithread capability.

Building blocks for big data

The data fabric powered by NetApp integrates data management services and applications (building blocks) for data access, control, protection, and security, as shown in the figure below.

The building blocks in the figure above include:

• **NetApp NFS direct access.** Provides the latest Hadoop and Spark clusters with direct access to NetApp NFS volumes without additional software or driver requirements.

• **NetApp Cloud Volumes ONTAP and Cloud Volume Services.** Software-defined connected storage based on ONTAP running in Amazon Web Services (AWS) or Azure NetApp Files (ANF) in Microsoft Azure cloud services.

• **NetApp SnapMirror technology.** Provides data protection capabilities between on-premises and ONTAP Cloud or NPS instances.

• **Cloud service providers.** These providers include AWS, Microsoft Azure, Google Cloud, and IBM Cloud.

• **PaaS.** Cloud-based analytics services such as Amazon Elastic MapReduce (EMR) and Databricks in AWS as well as Microsoft Azure HDInsight and Azure Databricks.

Hadoop data protection and NetApp

Hadoop DistCp is a native tool used for large intercluster and intracluster copying. The Hadoop DistCp basic process shown in the figure below is a typical backup workflow using Hadoop native tools such as MapReduce to copy Hadoop data from an HDFS source to a corresponding target.

The NetApp NFS direct access enables customers to set NFS as the target destination for the Hadoop DistCp tool to copy the data from HDFS source into an NFS share through MapReduce. The NetApp NFS direct
Overview of Hadoop data protection use cases

This section provides a high-level description of the data protection use cases, which constitute the focus of this paper. The remaining sections provide more details for each use case, such as the customer problem (scenario), requirements and challenges, and solutions.

Use case 1: Backing up Hadoop data

For this use case, NetApp NFS volume helped a large financial institution reduce the long backup window time from more than 24 hours to just under a few hours.

Use case 2: Backup and disaster recovery from the cloud to on-premises

By using the data fabric powered by NetApp as building blocks, a large broadcasting company was able to fulfill its requirement of backing up cloud data into its on-premise data center depending on the different modes of data transfers, such as on demand, instantaneous, or based on the Hadoop/Spark cluster load.

Use case 3: Enabling DevTest on existing Hadoop data

NetApp solutions helped an online music distributor to rapidly build multiple space-efficient Hadoop clusters in different branches to create reports and run daily DevTest tasks by using scheduled policies.

Use case 4: Data protection and multicloud connectivity

A large service provider used the data fabric powered by NetApp to provide multicloud analytics to its customers from different cloud instances.

Use case 5: Accelerate analytic workloads

One of the largest financial services and investment banks used the NetApp network-attached storage solution to reduce I/O wait time and accelerate its quantitative financial analytics platform.
Use case 1: Backing up Hadoop data

Scenario

In this scenario, the customer has a large on-premises Hadoop repository and wants to back it up for disaster recovery purposes. However, the customer’s current backup solution is costly and is suffering from a long backup window of more than 24 hours.

Requirements and challenges

The main requirements and challenges for this use case include:

- **Software backward compatibility:**
  - The proposed alternative backup solution should be compatible with the current running software versions used in the production Hadoop cluster.

- To meet the committed SLAs, the proposed alternative solution should achieve very low RPOs and RTOs.

- The backup created by the NetApp backup solution can be used in the Hadoop cluster built locally in the data center as well as the Hadoop cluster running in the disaster recovery location at the remote site.

- The proposed solution must be cost effective.

- The proposed solution must reduce the performance effect on the currently running, in-production analytics jobs during the backup times.

Customer’s existing backup solution

The figure below shows the original Hadoop native backup solution.

The production data is protected to tape through the intermediate backup cluster:

- HDFS1 data is copied to HDFS2 by running the `hadoop distcp -update <hdfs1> <hdfs2>` command.

- The backup cluster acts as an NFS gateway, and the data is manually copied to tape through the Linux `cp` command through the tape library.

The benefits of the original Hadoop native backup solution include:

- The solution is based on Hadoop native commands, which saves the user from having to learn new procedures.

- The solution leverages industry-standard architecture and hardware.

The disadvantages of the original Hadoop native backup solution include:
• The long backup window time exceeds 24 hours, which makes the production data vulnerable.
• Significant cluster performance degradation during backup times.
• Copying to tape is a manual process.
• The backup solution is expensive in terms of the hardware required and the human hours required for manual processes.

Backup solutions

Based on these challenges and requirements, and taking into consideration the existing backup system, three possible backup solutions were suggested. The following subsections describe each of these three different backup solutions, labeled solution A through solution C.

Solution A

In Solution A, the backup Hadoop cluster sends the secondary backups to NetApp NFS storage systems, eliminating the tape requirement, as shown in the figure below.

The detailed tasks for solution A include:

• The production Hadoop cluster has the customer’s analytics data in the HDFS that requires protection.
• The backup Hadoop cluster with HDFS acts as an intermediate location for the data. Just a bunch of disks (JBOD) provides the storage for HDFS in both the production and backup Hadoop clusters.
• Protect the Hadoop production data is protected from the production cluster HDFS to the backup cluster HDFS by running the `hadoop distcp -update -diff <hdfs1> <hdfs2>` command.

  The Hadoop snapshot is used to protect the data from production to the backup Hadoop cluster.

• The NetApp ONTAP storage controller provides an NFS exported volume, which is provisioned to the backup Hadoop cluster.

• By running the `hadoop distcp` command leveraging MapReduce and multiple mappers, the analytics data is protected from the backup Hadoop cluster to NFS.

After the data is stored in NFS on the NetApp storage system, NetApp Snapshot, SnapRestore, and
FlexClone technologies are used to back up, restore, and duplicate the Hadoop data as needed.

Hadoop data can be protected to the cloud as well as disaster recovery locations by using SnapMirror technology.

The benefits of solution A include:

- Hadoop production data is protected from the backup cluster.
- HDFS data is protected through NFS enabling protection to cloud and disaster recovery locations.
- Improves performance by offloading backup operations to the backup cluster.
- Eliminates manual tape operations
- Allows for enterprise management functions through NetApp tools.
- Requires minimal changes to the existing environment.
- Is a cost-effective solution.

The disadvantage of this solution is that it requires a backup cluster and additional mappers to improve performance.

The customer recently deployed solution A due to its simplicity, cost, and overall performance.

In this solution, SAN disks from ONTAP can be used instead of JBOD. This option offloads the backup cluster storage load to ONTAP; however, the downside is that SAN fabric switches are required.

**Solution B**

Solution B adds NFS volume to the production Hadoop cluster, which eliminates the need for the backup Hadoop cluster, as shown in the figure below.
The detailed tasks for solution B include:

- The NetApp ONTAP storage controller provisions the NFS export to the production Hadoop cluster.

The Hadoop native `hadoop distcp` command protects the Hadoop data from the production cluster HDFS to NFS.

- After the data is stored in NFS on the NetApp storage system, Snapshot, SnapRestore, and FlexClone technologies are used to back up, restore, and duplicate the Hadoop data as needed.

The benefits of solution B include:

- The production cluster is slightly modified for the backup solution, which simplifies implementation and reduces additional infrastructure cost.
- A backup cluster for the backup operation is not required.
- HDFS production data is protected in the conversion to NFS data.
- The solution allows for enterprise management functions through NetApp tools.

The disadvantage of this solution is that it's implemented in the production cluster, which can add additional administrator tasks in the production cluster.

Solution C

In solution C, the NetApp SAN volumes are directly provisioned to the Hadoop production cluster for HDFS storage, as shown in the figure below.
The detailed steps for solution C include:

- NetApp ONTAP SAN storage is provisioned at the production Hadoop cluster for HDFS data storage.
- NetApp Snapshot and SnapMirror technologies are used to back up the HDFS data from the production Hadoop cluster.
- There is no performance effect to production for the Hadoop/Spark cluster during the Snapshot copy backup process because the backup is at the storage layer.

Snapshot technology provides backups that complete in seconds regardless of the size of the data.

The benefits of solution C include:

- Space-efficient backup can be created by using Snapshot technology.
- Allows for enterprise management functions through NetApp tools.

Use case 2: Backup and disaster recovery from the cloud to on-premises

This use case is based on a broadcasting customer that needs to back up cloud-based analytics data to its on-premises data center, as illustrated in the figure below.
Scenario

In this scenario, the IoT sensor data is ingested into the cloud and analyzed by using an open source Apache Spark cluster within AWS. The requirement is to back up the processed data from the cloud to on-premises.

Requirements and challenges

The main requirements and challenges for this use case include:

- Enabling data protection should not cause any performance effect on the production Spark/Hadoop cluster in the cloud.
- Cloud sensor data needs to be moved and protected to on-premises in an efficient and secure way.
- Flexibility to transfer data from the cloud to on-premises under different conditions, such as on-demand, instantaneous, and during low-cluster load times.
**Solution**

The customer uses AWS Elastic Block Store (EBS) for its Spark cluster HDFS storage to receive and ingest data from remote sensors through Kafka. Consequently, the HDFS storage acts as the source for the backup data.

To fulfill these requirements, NetApp ONTAP Cloud is deployed in AWS, and an NFS share is created to act as the backup target for the Spark/Hadoop cluster.

After the NFS share is created, copy the data from the HDFS EBS storage into the ONTAP NFS share. After the data resides in NFS in ONTAP Cloud, SnapMirror technology can be used to mirror the data from the cloud into on-premises storage as needed in a secure and efficient way.

This image shows the backup and disaster recovery from cloud to on-premises solution.

**Use case 3: Enabling DevTest on existing Hadoop data**

In this use case, the customer’s requirement is to rapidly and efficiently build new Hadoop/Spark clusters based on an existing Hadoop cluster containing a large amount of analytics data for DevTest and reporting purposes in the same data center as well as remote locations.

**Scenario**

In this scenario, multiple Spark/Hadoop clusters are built from a large Hadoop data lake implementation on-premises as well as at disaster recovery locations.

**Requirements and challenges**

The main requirements and challenges for this use case include:

- Create multiple Hadoop clusters for DevTest, QA, or any other purpose that requires access to the same production data. The challenge here is to clone a very large Hadoop cluster multiple times instantaneously and in a very space-efficient manner.
• Sync the Hadoop data to DevTest and reporting teams for operational efficiency.
• Distribute the Hadoop data by using the same credentials across production and new clusters.
• Use scheduled policies to efficiently create QA clusters without affecting the production cluster.

Solution

FlexClone technology is used to answer the requirements just described. FlexClone technology is the read/write copy of a Snapshot copy. It reads the data from parent Snapshot copy data and only consumes additional space for new/modified blocks. It is fast and space-efficient.

First, a Snapshot copy of the existing cluster was created by using a NetApp consistency group. Snapshot copies within NetApp System Manager or the storage admin prompt. The consistency group Snapshot copies are application-consistent group Snapshot copies, and the FlexClone volume is created based on consistency group Snapshot copies. It is worth mentioning that a FlexClone volume inherits the parent volume’s NFS export policy. After the Snapshot copy is created, a new Hadoop cluster must be installed for DevTest and reporting purposes, as shown in the figure below. The cloned NFS volume from the new Hadoop cluster access the the NFS data.

This image shows the Hadoop cluster for DevTest.

Use case 4: Data protection and multicloud connectivity

This use case is relevant for a cloud service partner tasked with providing multicloud connectivity for customers' big data analytics data.
Scenario

In this scenario, IoT data received in AWS from different sources is stored in a central location in NPS. The NPS storage is connected to Spark/Hadoop clusters located in AWS and Azure enabling big data analytics applications running in multiple clouds accessing the same data.

Requirements and challenges

The main requirements and challenges for this use case include:

• Customers want to run analytics jobs on the same data using multiple clouds.
• Data must be received from different sources such as on-premises and cloud through different sensors and hubs.
• The solution must be efficient and cost-effective.
• The main challenge is to build a cost-effective and efficient solution that delivers hybrid analytics services between on-premises and across different clouds.

Solution

This image illustrates the data protection and multicloud connectivity solution.

As shown in the figure above, data from sensors is streamed and ingested into the AWS Spark cluster through Kafka. The data is stored in an NFS share residing in NPS, which is located outside of the cloud provider within an Equinix data center. Because NetApp NPS is connected to Amazon AWS and Microsoft Azure through Direct Connect and Express Route connections, respectively, customers can access the NFS data from both Amazon and AWS analytics clusters. This approach solves having cloud analytics across multiple hyperscalers.

Consequently, because both on-premises and NPS storage runs ONTAP software, SnapMirror can mirror the NPS data into the on-premises cluster, providing hybrid cloud analytics across on-premises and multiple clouds.

For the best performance, NetApp typically recommends using multiple network interfaces and direct connection/express routes to access the data from cloud instances.
Use case 5: Accelerate analytic workloads

In this scenario, a large financial services and investment bank’s analytics platform was modernized using the NetApp NFS storage solution to achieve significant improvement in analyzing investment risks and derivatives for its asset management and quantitative business unit.

**Scenario**

In the customer’s existing environment, the Hadoop infrastructure used for the analytics platform leveraged internal storage from the Hadoop servers. Due to proprietary nature of JBOD environment, many internal customers within the organization were unable to take advantage of their Monte Carlo quantitative model, a simulation that relies on the recurring samples of real-time data. The suboptimal ability to understand the effects of uncertainty in market movements was serving unfavorably for the quantitative asset management business unit.

**Requirements and challenges**

The quantitative business unit at the bank wanted an efficient forecasting method to attain accurate and timely predictions. To do so, the team recognized the need to modernize the infrastructure, reduce existing I/O wait time and improve performance on the analytic applications such as Hadoop and Spark to efficiently simulate investment models, measure potential gains and analyze risks.

**Solution**

The customer had JBOD for their existing Spark solution. NetApp ONTAP, NetApp StorageGRID, and MinIO Gateway to NFS was then leveraged to reduce the I/O wait time for the bank’s quantitative finance group that runs simulation and analysis on investment models that assess potential gains and risks. This image shows the Spark solution with NetApp storage.
As shown in figure above, AFF A800, A700 systems, and StorageGRID were deployed to access parquet files through NFS and S3 protocols in a six-node Hadoop cluster with Spark, and YARN and Hive metadata services for data analytic operations.

A direct-attached storage (DAS) solution in the customer's old environment had the disadvantage to scale compute and storage independently. With NetApp ONTAP solution for Spark, the bank’s financial analytics business unit was able to decouple storage from compute and seamlessly bring infrastructure resources more effectively as needed.

By using ONTAP with NFS, the compute server CPUs were almost fully utilized for Spark SQL jobs and the I/O wait time was reduced by nearly 70%, therefore providing better compute power and performance boost to Spark workloads. Subsequently, increasing CPU utilization also enabled the customer to leverage GPUs, such as GPUDirect, for further platform modernization. Additionally, StorageGRID provides a low-cost storage option for Spark workloads and MinIO Gateway provides secure access to NFS data through the S3 protocol. For data in the cloud, NetApp recommends Cloud Volumes ONTAP, Azure NetApp Files, and NetApp Cloud Volumes Service.

**Conclusion**

This section provides a summary of the use cases and solutions provided by NetApp to fulfill various Hadoop data protection requirements. By using the data fabric powered by
NetApp, customers can:

- Have the flexibility to choose the right data protection solutions by leveraging NetApp’s rich data management capabilities and integration with Hadoop native workflows.
- Reduce their Hadoop cluster backup window time by almost 70%.
- Eliminate any performance effect resulting from Hadoop cluster backups.
- Provide multicloud data protection and data access from different cloud providers simultaneously to a single source of analytics data.
- Create fast and space-efficient Hadoop cluster copies by using FlexClone technology.

Where to find additional information

To learn more about the information described in this document, see the following documents and/or websites:

- NetApp Big Data Analytics Solutions
- Apache Spark Workload with NetApp Storage
- NetApp Storage Solutions for Apache Spark
  https://www.netapp.com/media/16864-tr-4570.pdf
- Apache Hadoop on data fabric enabled by NetApp
  https://www.netapp.com/media/16877-tr-4529.pdf

Acknowledgements

- Paul Burland, Sales Rep, ANZ Victoria District Sales, NetApp
- Hoseb Dermanilian, Business Development Manager, NetApp
- Lee Dorrier, Director MPSG, NetApp
- David Thiessen, Systems Engineer, ANZ Victoria District SE, NetApp

Version history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Document version history</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 1.0</td>
<td>January 2018</td>
<td>Initial release</td>
</tr>
<tr>
<td>Version 2.0</td>
<td>October 2021</td>
<td>Updated with use case #5: Accelerate analytic workload</td>
</tr>
<tr>
<td>Version 3.0</td>
<td>November 2023</td>
<td>Removed NIPAM details</td>
</tr>
</tbody>
</table>
Modern data analytics - Different solutions for different analytics strategies

This white paper describes NetApp modern data analytics solution strategies. It includes details about the business outcomes, customer challenges, technology trends, competition legacy architecture, modern workflows, use cases, industries, cloud, technology partners, data movers, NetApp Active IQ, NetApp DataOps Toolkit, Hadoop to Spark, software-defined storage with NetApp Astra Control, containers, enterprise data management, archiving, and tiering toward achieving the goals of AI and analytics and how NetApp and customers together are modernizing their data architecture.

TR-4623: NetApp E-Series E5700 and Splunk Enterprise

Mitch Blackburn, NetApp

TR-4623 describes the integrated architecture of the NetApp E-Series and Splunk design. Optimized for node storage balance, reliability, performance, storage capacity, and density, this design employs the Splunk clustered index node model, with higher scalability and lower TCO. Decoupling storage from compute provides the ability to scale each separately, saving the cost of overprovisioning one or the other. In addition, this document summarizes the performance test results obtained from a Splunk machine log event simulation tool.

NVA-1157-DEPLOY: Apache Spark workload with NetApp storage solution

Karthikeyan Nagalingam, NetApp

NVA-1157-DEPLOY describes the performance and functionality validation of Apache Spark SQL on NetApp NFS AFF storage systems. It reviews the configuration, architecture, and performance testing based on various scenarios, as well as recommendations for using Spark with NetApp ONTAP data management software. It also covers test results based on just a bunch of disks (JBOD) versus the NetApp AFF A800 storage controller.
NetApp Hybrid Multicloud with VMware Solutions

VMware for Public Cloud

Overview of NetApp Hybrid Multicloud with VMware

Most IT organizations follow the hybrid cloud-first approach. These organizations are in a transformation phase and customers are evaluating their current IT landscape and then migrating their workloads to the cloud based on the assessment and discovery exercise.

The factors for customers migrating to the cloud can include elasticity and burst, data center exit, data center consolidation, end-of-life scenarios, mergers, acquisitions, and so on. The reason for this migration can vary based on each organization and their respective business priorities. When moving to the hybrid cloud, choosing the right storage in the cloud is very important in order to unleash the power of cloud deployment and elasticity.

VMware Cloud options in Public Cloud

This section describes how each of the cloud providers support a VMware Software Defined Data Center (SDDC) and/or VMware Cloud Foundation (VCF) stack within their respective public cloud offerings.

Azure VMware Solution

Azure VMware Solution is a hybrid cloud service that allows for fully functioning VMware SDDCs within the Microsoft Azure public cloud. Azure VMware Solution is a first-party solution fully managed and supported by Microsoft, verified by VMware leveraging Azure infrastructure. This means that when Azure VMware Solution is deployed, customer’s get VMware’s ESXi for compute virtualization, vSAN for hyper-converged storage, and NSX for networking and security, all while taking advantage of Microsoft Azure’s global presence, class-leading data center facilities and proximity to the rich ecosystem of native Azure services and solutions.

VMware Cloud on AWS

VMware Cloud on AWS brings VMware’s enterprise-class SDDC software to the AWS Cloud with optimized access to native AWS services. Powered by VMware Cloud Foundation, VMware Cloud on AWS integrates VMware’s compute, storage, and network virtualization products (VMware vSphere, VMware vSAN, and VMware NSX) along with VMware vCenter Server management, optimized to run on dedicated, elastic, bare-metal AWS infrastructure.

Google Cloud VMware Engine
Google Cloud VMware Engine is an infrastructure-as-a-service (IaaS) offering built on Google Cloud’s highly performant scalable infrastructure and VMware Cloud Foundation stack – VMware vSphere, vCenter, vSAN, and NSX-T. This service enables a fast path to the cloud, seamlessly migrating or extending existing VMware workloads from on-premises environments to Google Cloud Platform without the cost, effort, or risk of rearchitecting applications or retooling operations. It is a service sold and supported by Google, working closely with VMware.

SDDC private cloud and NetApp Cloud Volumes colocation provides the best performance with minimal network latency.

Did you know?

Regardless of the cloud used, when a VMware SDDC is deployed, the initial cluster includes the following products:

- VMware ESXi hosts for compute virtualization with a vCenter Server appliance for management
- VMware vSAN hyper-converged storage incorporating the physical storage assets of each ESXi host
- VMware NSX for virtual networking and security with an NSX Manager cluster for management

Storage configuration

For customers planning to host storage-intensive workloads and scale out on any cloud-hosted VMware solution, the default hyper-converged infrastructure dictates that the expansion should be on both the compute and storage resources.

By integrating with NetApp Cloud Volumes, such as Azure NetApp Files, Amazon FSx for NetApp ONTAP, Cloud Volumes ONTAP (available in all three major hyperscalers), and Cloud Volumes Service for Google Cloud, customers now have options to independently scale their storage separately, and only add compute nodes to the SDDC cluster as needed.

Notes:

- VMware does not recommend unbalanced cluster configurations, hence expanding storage means adding more hosts, which implies more TCO.
- Only one vSAN environment is possible. Therefore, all storage traffic will compete directly with production workloads.
- There is no option to provide multiple performance tiers to align application requirements, performance, and cost.
- It is very easy to reach the limits of storage capacity of vSAN built on top of the cluster hosts. Use NetApp Cloud Volumes to scale storage to either host active datasets or tier cooler data to persistent storage.

Azure NetApp Files, Amazon FSx for NetApp ONTAP, Cloud Volumes ONTAP (available in all three major hyperscalers), and Cloud Volumes Service for Google Cloud can be used in conjunction with guest VMs. This hybrid storage architecture consists of a vSAN datastore that holds the guest operating system and application binary data. The application data is attached to the VM through a guest-based iSCSI initiator or the NFS/SMB mounts that communicate directly with Amazon FSx for NetApp ONTAP, Cloud Volume ONTAP, Azure NetApp Files and Cloud Volumes Service for Google Cloud respectively. This configuration allows you to easily overcome challenges with storage capacity as with vSAN, the available free space depends on the slack space and storage policies used.

Let’s consider a three-node SDDC cluster on VMware Cloud on AWS:
The total raw capacity for a three-node SDDC = 31.1TB (roughly 10TB for each node).
The slack space to be maintained before additional hosts are added = 25% = (.25 x 31.1TB) = 7.7TB.
The usable raw capacity after slack space deduction = 23.4TB
The effective free space available depends on the storage policy applied.

For example:

- RAID 0 = effective free space = 23.4TB (usable raw capacity/1)
- RAID 1 = effective free space = 11.7TB (usable raw capacity/2)
- RAID 5 = effective free space = 17.5TB (usable raw capacity/1.33)

Thus, using NetApp Cloud Volumes as guest-connected storage would help in expanding the storage and optimizing the TCO while meeting the performance and data protection requirements.

In-guest storage was the only available option at the time this document was written. As supplemental NFS datastore support becomes available, additional documentation will be available here.

Points to Remember

- In hybrid storage models, place tier 1 or high priority workloads on vSAN datastore to address any specific latency requirements because they are part of the host itself and within proximity. Use in-guest mechanisms for any workload VMs for which transactional latencies are acceptable.

- Use NetApp SnapMirror® technology to replicate the workload data from the on-premises ONTAP system to Cloud Volumes ONTAP or Amazon FSx for NetApp ONTAP to ease migration using block-level mechanisms. This does not apply to Azure NetApp Files and Cloud Volumes Services. For migrating data to Azure NetApp Files or Cloud Volumes Services, use NetApp XCP, BlueXP Copy and Sync, rysnc or robocopy depending on the file protocol used.

- Testing shows 2-4ms additional latency while accessing storage from the respective SDDCs. Factor this additional latency into the application requirements when mapping the storage.

- For mounting guest-connected storage during test failover and actual failover, make sure iSCSI initiators are reconfigured, DNS is updated for SMB shares, and NFS mount points are updated in fstab.

- Make sure that in-guest Microsoft Multipath I/O (MPIO), firewall, and disk timeout registry settings are configured properly inside the VM.

This applies to guest connected storage only.

Benefits of NetApp cloud storage

NetApp cloud storage offers the following benefits:

- Improves compute-to-storage density by scaling storage independently of compute.
- Allows you to reduce the host count, thus reducing the overall TCO.
- Compute node failure does not impact storage performance.
- The volume reshaping and dynamic service-level capability of Azure NetApp Files allows you to optimize cost by sizing for steady-state workloads, and thus preventing over provisioning.
- The storage efficiencies, cloud tiering, and instance-type modification capabilities of Cloud Volumes ONTAP allow optimal ways of adding and scaling storage.
• Prevents over provisioning storage resources are added only when needed.
• Efficient Snapshot copies and clones allow you to rapidly create copies without any performance impact.
• Helps address ransomware attacks by using quick recovery from Snapshot copies.
• Provides efficient incremental block transfer-based regional disaster recovery and integrated backup block level across regions provides better RPO and RTOs.

Assumptions

• SnapMirror technology or other relevant data migration mechanisms are enabled. There are many connectivity options, from on-premises to any hyperscaler cloud. Use the appropriate path and work with the relevant networking teams.

• In-guest storage was the only available option at the time this document was written. As supplemental NFS datastore support becomes available, additional documentation will be available here.

Engage NetApp solution architects and respective hyperscaler cloud architects for planning and sizing of storage and the required number of hosts. NetApp recommends identifying the storage performance requirements before using the Cloud Volumes ONTAP sizer to finalize the storage instance type or the appropriate service level with the right throughput.

Detailed architecture

From a high-level perspective, this architecture (shown in the figure below) covers how to achieve hybrid Multicloud connectivity and app portability across multiple cloud providers using NetApp Cloud Volumes ONTAP, Cloud Volumes Service for Google Cloud and Azure NetApp Files as an additional in-guest storage option.
NetApp Solutions for VMware in Hyperscalers

Learn more about the capabilities that NetApp brings to the three (3) primary hyperscalers - from NetApp as a guest connected storage device or a supplemental NFS datastore to migrating workflows, extending/bursting to the cloud, backup/restore and disaster recovery.

Pick your cloud and let NetApp do the rest!

To see the capabilities for a specific hyperscaler, click on the appropriate tab for that hyperscaler.

Jump to the section for the desired content by selecting from the following options:

- VMware in the Hyperscalers Configuration
- NetApp Storage Options
- NetApp / VMware Cloud Solutions

VMware in the Hyperscalers Configuration

As with on-premises, planning a cloud based virtualization environment is critical for a successful production-ready environment for creating VMs and migration.
AWS / VMC
This section describes how to set up and manage VMware Cloud on AWS SDDC and use it in combination with the available options for connecting NetApp storage.

In-guest storage is the only supported method of connecting Cloud Volumes ONTAP to AWS VMC.

The setup process can be broken down into the following steps:

• Deploy and Configure VMware Cloud for AWS
• Connect VMware Cloud to FSx ONTAP

View the detailed configuration steps for VMC.

Azure / AVS
This section describes how to set up and manage Azure VMware Solution and use it in combination with the available options for connecting NetApp storage.

In-guest storage is the only supported method of connecting Cloud Volumes ONTAP to Azure VMware Solution.

The setup process can be broken down into the following steps:

• Register the resource provider and create a private cloud
• Connect to a new or existing ExpressRoute virtual network gateway
• Validate the network connectivity and access the private cloud

View the detailed configuration steps for AVS.

GCP / GCVE
This section describes how to set up and manage GCVE and use it in combination with the available options for connecting NetApp storage.

In-guest storage is the only supported method of connecting Cloud Volumes ONTAP and Cloud Volumes Services to GCVE.

The setup process can be broken down into the following steps:

• Deploy and Configure GCVE
• Enable Private Access to GCVE

View the detailed configuration steps for GCVE.

NetApp Storage Options
NetApp storage can be utilized in several ways - either as guest connected or as a supplemental NFS datastore - within each of the 3 major hyperscalers.

Please visit Supported NetApp Storage Options for more information.
**AWS / VMC**
AWS supports NetApp storage in the following configurations:

- FSx ONTAP as guest connected storage
- Cloud Volumes ONTAP (CVO) as guest connected storage
- FSx ONTAP as a supplemental NFS datastore

View the detailed guest connect storage options for VMC.
View the detailed supplemental NFS datastore options for VMC.

**Azure / AVS**
Azure supports NetApp storage in the following configurations:

- Azure NetApp Files (ANF) as guest connected storage
- Cloud Volumes ONTAP (CVO) as guest connected storage
- Azure NetApp Files (ANF) as a supplemental NFS datastore

View the detailed guest connect storage options for AVS.
View the detailed supplemental NFS datastore options for AVS.

**GCP / GCVE**
Google Cloud supports NetApp storage in the following configurations:

- Cloud Volumes ONTAP (CVO) as guest connected storage
- Cloud Volumes Service (CVS) as guest connected storage
- Cloud Volumes Service (CVS) as a supplemental NFS datastore

View the detailed guest connect storage options for GCVE.

Read more about NetApp Cloud Volumes Service datastore support for Google Cloud VMware Engine (NetApp blog) or How to use NetApp CVS as datastores for Google Cloud VMware Engine (Google blog)

**NetApp / VMware Cloud Solutions**
With NetApp and VMware cloud solutions, many use cases are simple to deploy in your hyperscaler of choice. VMware defines the primary cloud workload use-cases as:

- Protect (includes both Disaster Recovery and Backup / Restore)
- Migrate
- Extend
**Supported Configurations for NetApp Hybrid Multicloud with VMware**

Understanding the combinations for NetApp storage support in the major hyperscalers.

<table>
<thead>
<tr>
<th></th>
<th>Guest Connected</th>
<th>Supplemental NFS Datastore</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AWS</strong></td>
<td>CVO</td>
<td>FSx ONTAP</td>
</tr>
<tr>
<td></td>
<td>FSx ONTAP ONTAP</td>
<td>Details</td>
</tr>
<tr>
<td></td>
<td>CVO</td>
<td>FSx ONTAP</td>
</tr>
<tr>
<td></td>
<td>ANF Details</td>
<td>ANF</td>
</tr>
<tr>
<td><strong>GCP</strong></td>
<td>CVO</td>
<td>CVS</td>
</tr>
<tr>
<td></td>
<td>CVS Details</td>
<td>CVS</td>
</tr>
</tbody>
</table>

**Configuring the virtualization environment in the cloud provider**

Details for how to configure the virtualization environment in each of the supported hyperscalers are covered here.
This section describes how to set up and manage VMware Cloud on AWS SDDC and use it in combination with the available options for connecting NetApp storage.

In-guest storage is the only supported method of connecting Cloud Volumes ONTAP to AWS VMC.

The setup process can be broken down into the following steps:

• Deploy and Configure VMware Cloud for AWS
• Connect VMware Cloud to FSx ONTAP

View the detailed configuration steps for VMC.

In-guest storage is the only supported method of connecting Cloud Volumes ONTAP to Azure VMware Solution.

The setup process can be broken down into the following steps:

• Register the resource provider and create a private cloud
• Connect to a new or existing ExpressRoute virtual network gateway
• Validate the network connectivity and access the private cloud

View the detailed configuration steps for AVS.

In-guest storage is the only supported method of connecting Cloud Volumes ONTAP and Cloud Volumes Services to GCVE.

The setup process can be broken down into the following steps:

• Deploy and Configure GCVE
• Enable Private Access to GCVE

View the detailed configuration steps for GCVE.

Deploy and configure the Virtualization Environment on AWS

As with on-premises, planning VMware Cloud on AWS is critical for a successful production-ready environment for creating VMs and migration.

This section describes how to set up and manage VMware Cloud on AWS SDDC and use it in combination
with the available options for connecting NetApp storage.

In-guest storage is currently the only supported method of connecting Cloud Volumes ONTAP (CVO) to AWS VMC.

The setup process can be broken down into the following steps:
Deploy and configure VMware Cloud for AWS

VMware Cloud on AWS provides for a cloud native experience for VMware based workloads in the AWS ecosystem. Each VMware Software-Defined Data Center (SDDC) runs in an Amazon Virtual Private Cloud (VPC) and provides a full VMware stack (including vCenter Server), NSX-T software-defined networking, vSAN software-defined storage, and one or more ESXi hosts that provide compute and storage resources to your workloads.

This section describes how to set up and manage VMware Cloud on AWS and use it in combination with Amazon FSx for NetApp ONTAP and/or Cloud Volumes ONTAP on AWS with in-guest storage.

In-guest storage is currently the only supported method of connecting Cloud Volumes ONTAP (CVO) to AWS VMC.

The setup process can be broken down into three parts:

Register for an AWS Account

Register for an Amazon Web Services Account.

You need an AWS account to get started, assuming there isn't one created already. New or existing, you need administrative privileges in the account for many steps in this procedure. See this link for more information regarding AWS credentials.

Register for a My VMware Account

Register for a My VMware account.

For access to VMware’s cloud portfolio (including VMware Cloud on AWS), you need a VMware customer account or a My VMware account. If you have not already done so, create a VMware account here.
Provision SDDC in VMware Cloud

After the VMware account is configured and proper sizing is performed, deploying a Software-Defined Data Center is the obvious next step for using the VMware Cloud on AWS service. To create an SDDC, pick an AWS region to host it, give the SDDC a name, and specify how many ESXi hosts you want the SDDC to contain. If you don’t already have an AWS account, you can still create a starter configuration SDDC that contains a single ESXi host.

1. Log into the VMware Cloud Console using your existing or newly created VMware credentials.

![VMware Cloud Console](image)

2. Configure the AWS region, deployment, and host type and the SDDC name:

![SDDC Configuration](image)

3. Connect to the desired AWS account and execute the AWS Cloud Formation stack.
Quick create stack

Template

Template URL:
/modular-cloud/mgmt/cloudformation/stacks/cloudformation-stack-template

Stack description:
This template is created by VMware Cloud on AWS for SDDC deployment and maintenance. Please do not remove.

Stack name

Stack name:

Stack name: svmware-cloud-formation-8376555555-8324-8376-8324-8376-8324-8376

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

Capabilities:

The following resources require capabilities (AWS IAM roles):

This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Choose the role you want to create or update these resources and that they have the minimum required permissions. Learn more.

I acknowledge that AWS CloudFormation might create IAM resource.
Single-host configuration is used in this validation.

4. Select the desired AWS VPC to connect the VMC environment with.
5. Configure the VMC Management Subnet; this subnet contains VMC-managed services like vCenter, NSX, and so on. Do not choose an overlapping address space with any other networks that need connectivity to the SDDC environment. Finally, follow the recommendations for CIDR size notated below.

6. Review and acknowledge the SDDC configuration, and then click deploy the SDDC.
The deployment process typically takes approximately two hours to complete.

7. After completion, the SDDC is ready for use.
For a step-by-step guide on SDDC deployment, see Deploy an SDDC from the VMC Console.
To connect VMware Cloud to FSx ONTAP, complete the following steps:

1. With VMware Cloud deployment completed and connected to AWS VPC, you must deploy Amazon FSx for NetApp ONTAP into a new VPC rather than the original connected VPC (see the screenshot below). FSx (NFS and SMB floating IPs) is not accessible if it is deployed in the connected VPC. Keep in mind that ISCSI endpoints like Cloud Volumes ONTAP work just fine from the connected VPC.

2. Deploy an additional VPC in the same region, and then deploy Amazon FSx for NetApp ONTAP into the new VPC.

Configuration of an SDDC group in the VMware Cloud console enables the networking configuration options required to connect to the new VPC where FSx is deployed. In step 3, verify that “Configuring VMware Transit Connect for your group will incur charges per attachment and data transfers” is checked, and then choose Create Group. The process can take a few minutes to complete.
### Create SDDC Group

**1. Name and Description:** Create a name and description for your group.

- **Name:** 
- **Description:** SDDC_group

**2. Membership:** Members: 1

**3. Acknowledgement:**

- Please confirm that you are aware of the following before creating the SDDC Group:
  - Creating VMware Transit Connect for your group will incur charges per attachment and data transfers.
  - Create shared rules to establish connectivity between the SDDCs in the group.

**CREATE GROUP**
3. Attach the newly created VPC to the just created SDDC group. Select the External VPC tab and follow the instructions for attaching an External VPC to the group. This process can take 10 to 15 minutes to complete.
4. As part of the external VPC process, you are prompted through the AWS console to a new shared resource via the Resource Access Manager. The shared resource is the AWS Transit Gateway managed by VMware Transit Connect.
5. Create the Transit Gateway Attachment.

6. Back on the VMC Console, Accept the VPC attachment. This process can take approximately 10 minutes to complete.
7. While in the External VPC tab, click the edit icon in the Routes column and add in the following required routes:

- A route for the floating IP range for Amazon FSx for NetApp ONTAP floating IPs.
- A route for the floating IP range for Cloud Volumes ONTAP (if applicable).
- A route for the newly created external VPC address space.

8. Finally, allow bidirectional traffic firewall rules for access to FSx/CVO. Follow these detailed steps for compute gateway firewall rules for SDDC workload connectivity.
9. After the firewall groups are configured for both the Management and Compute gateway, the vCenter can be accessed as follows:

The next step is to verify that Amazon FSx ONTAP or Cloud Volumes ONTAP is configured depending on your requirements and that the volumes are provisioned to offload storage components from vSAN to optimize the deployment.
Deploy and configure the Virtualization Environment on Azure

As with on-premises, planning Azure VMware Solution is critical for a successful production-ready environment for creating VMs and migration.

This section describes how to set up and manage Azure VMware Solution and use it in combination with the available options for connecting NetApp storage.

The setup process can be broken down into the following steps:
Register the resource provider and create a private cloud

To use Azure VMware Solution, first register the resource provider within the identified subscription:

1. Sign in to the Azure portal.
2. On the Azure portal menu, select All Services.
3. In the All Services dialog box, enter the subscription and then select Subscriptions.
4. To view, select the subscription from the subscription list.
5. Select Resource Providers and enter Microsoft.AVS into the search.
6. If the resource provider is not registered, select Register.
7. After the resource provider is registered, create an Azure VMware Solution private cloud by using the Azure portal.

8. Sign in to the Azure portal.


10. In the Search the Marketplace text box, enter Azure VMware Solution and select it from the results.


12. From the Basics tab, enter the values in the fields and select Review + Create.

Notes:

• For a quick start, gather the required information during the planning phase.

• Select an existing resource group or create a new resource group for the private cloud. A resource group is a logical container in which the Azure resources are deployed and managed.

• Make sure the CIDR address is unique and does not overlap with other Azure Virtual Networks or on-premises networks. The CIDR represents the private cloud management network and is used for the cluster management services, such as vCenter Server and NSX-T Manager. NetApp recommends using a /22 address space. In this example, 10.21.0.0/22 is used.
The provisioning process takes approximately 4–5 hours. After the process is complete, verify that the deployment was successful by accessing the private cloud from the Azure portal. A status of Succeeded is displayed when the deployment is complete.

An Azure VMware Solution private cloud requires an Azure Virtual Network. Because Azure VMware Solution doesn’t support on-premises vCenter, additional steps are required to integrate with an existing on-premises environment. Setting up an ExpressRoute circuit and a virtual network gateway is also required. While waiting for the cluster provisioning to complete, create a new virtual network or use an existing one to connect to Azure VMware Solution.
Connect to a new or existing ExpressRoute virtual network gateway

To create a new Azure Virtual Network (VNet), select the Azure VNet Connect tab. Alternatively, you can create one manually from the Azure portal by using the Create Virtual Network wizard:

1. Go to Azure VMware Solution private cloud and access Connectivity under the Manage option.
2. Select Azure VNet Connect.
3. To create a new VNet, select the Create New option.

This feature allows a VNet to be connected to the Azure VMware Solution private cloud. The VNet enables communication between workloads in this virtual network by automatically creating required components (for example, jump box, shared services such as Azure NetApp Files, and Cloud Volume ONTAP) to the private cloud created in Azure VMware Solution over ExpressRoute.

**Note:** The VNet address space should not overlap with the private cloud CIDR.

4. Provide or update the information for the new VNet and select OK.
The VNet with the provided address range and gateway subnet is created in the designated subscription and resource group.

If you create a VNet manually, create a virtual network gateway with the appropriate SKU and ExpressRoute as the gateway type. After the deployment is complete, connect the ExpressRoute connection to the virtual network gateway containing Azure VMware Solution private cloud using the authorization key. For more information, see Configure networking for your VMware private cloud in Azure.
Validate the network connect and access to Azure VMware Solution private cloud

Azure VMware Solution does not allow you to manage a private cloud with on-premises VMware vCenter. Instead, jump host is required to connect to the Azure VMware Solution vCenter instance. Create a jump host in the designated resource group and sign in to the Azure VMware Solution vCenter. This jump host should be a Windows VM on the same virtual network that was created for connectivity and should provide access to both vCenter and the NSX Manager.

Create a virtual machine

[Image of virtual machine creation interface]

After the virtual machine is provisioned, use the Connect option to access RDP.
Sign in to vCenter from this newly created jump host virtual machine by using the cloud admin user. To access the credentials, go to the Azure portal and navigate to Identity (under the Manage option within the private cloud). The URLs and user credentials for the private cloud vCenter and NSX-T Manager can be copied from here.

In the Windows virtual machine, open a browser and navigate to the vCenter web client URL (https://10.21.0.2/) and use the admin user name as cloudadmin@vsphere.local and paste the copied password. Similarly, NSX-T manager can also be accessed using the web client URL (https://10.21.0.3/) and use the admin user name and paste the copied password to create new segments or modify the existing tier gateways.

The web client URLs are different for each SDDC provisioned.
The Azure VMware Solution SDDC is now deployed and configured. Leverage ExpressRoute Global Reach to connect the on-premises environment to Azure VMware Solution private cloud. For more information, see Peer on-premises environments to Azure VMware Solution.

**Deploy and configure the Virtualization Environment on Google Cloud Platform (GCP)**

As with on-premises, planning Google Cloud VMware Engine (GCVE) is critical for a successful production-ready environment for creating VMs and migration.

This section describes how to set up and manage GCVE and use it in combination with the available options for connecting NetApp storage.

The setup process can be broken down into the following steps:
Deploy and configure GCVE

To configure a GCVE environment on GCP, login to the GCP console and access the VMware Engine portal.

Click on the “New Private Cloud” button and enter the desired configuration for the GCVE Private Cloud. On “Location”, make sure to deploy the private cloud in the same Region/Zone where CVS/CVO is deployed, to ensure the best performance and lowest latency.

Pre-requisites:

- Setup VMware Engine Service Admin IAM role
- Enable VMWare Engine API access and node quota
- Make sure that the CIDR range doesn’t overlap with any of your on-premises or cloud subnets. The CIDR range must be /27 or higher.

Note: Private cloud creation can take between 30 minutes to 2 hours.
Enable Private Access to GCVE

Once the Private Cloud is provisioned, configure private access to the Private Cloud for high-throughput and low-latency data-path connection.

This will ensure that the VPC network where Cloud Volumes ONTAP instances are running is able to communicate with the GCVE Private Cloud. To do so, follow the GCP documentation. For the Cloud Volume Service, establish a connection between VMware Engine and Cloud Volumes Service by performing a one-time peering between the tenant host projects. For detailed steps, follow this link.

Sign in to vcenter using the CloudOwner@gve.local user. To access the credentials, go to the VMware Engine portal, Go to Resources, and select the appropriate private cloud. In the Basic info section, click the View link for either vCenter login info (vCenter Server, HCX Manager) or NSX-T login info (NSX Manager).

In a Windows virtual machine, open a browser and navigate to the vCenter web client URL (https://10.0.16.6/) and use the admin user name as CloudOwner@gve.local and paste the copied password. Similarly, NSX-T manager can also be accessed using the web client URL (https://10.0.16.11/) and use the admin user name and paste the copied password to create new segments or modify the existing tier gateways.

For connecting from an on-premises network to VMware Engine private cloud, leverage cloud VPN or Cloud Interconnect for appropriate connectivity and make sure the required ports are open. For detailed steps, follow this link.
Deploy NetApp Cloud Volume Service supplemental datastore to GCVE

Refer Procedure to deploy supplemental NFS datastore with NetApp CVS to GCVE

NetApp Storage options for Public Cloud Providers

Explore the options for NetApp as storage in the three major hyperscalers.
AWS / VMC
AWS supports NetApp storage in the following configurations:

- FSx ONTAP as guest connected storage
- Cloud Volumes ONTAP (CVO) as guest connected storage
- FSx ONTAP as a supplemental NFS datastore

View the detailed guest connect storage options for VMC.
View the detailed supplemental NFS datastore options for VMC.

Azure / AVS
Azure supports NetApp storage in the following configurations:

- Azure NetApp Files (ANF) as guest connected storage
- Cloud Volumes ONTAP (CVO) as guest connected storage
- Azure NetApp Files (ANF) as a supplemental NFS datastore

View the detailed guest connect storage options for AVS.
View the detailed supplemental NFS datastore options for AVS.

GCP / GCVE
Google Cloud supports NetApp storage in the following configurations:

- Cloud Volumes ONTAP (CVO) as guest connected storage
- Cloud Volumes Service (CVS) as guest connected storage
- Cloud Volumes Service (CVS) as a supplemental NFS datastore

View the detailed guest connect storage options for GCVE.

Read more about NetApp Cloud Volumes Service datastore support for Google Cloud VMware Engine (NetApp blog) or How to use NetApp CVS as datastores for Google Cloud VMware Engine (Google blog)

TR-4938: Mount Amazon FSx for ONTAP as a NFS datastore with VMware Cloud on AWS

Niyaz Mohamed, NetApp

Introduction

Every successful organization is on a path of transformation and modernization. As part of this process, companies typically use their existing VMware investments to leverage cloud benefits and exploring how to migrate, burst, extend, and provide disaster recovery for processes as seamlessly as possible. Customers migrating to the cloud must evaluate the use cases for elasticity and burst, data-center exit, data-center consolidation, end-of-life scenarios, mergers, acquisitions, and so on.

Although VMware Cloud on AWS is the preferred option for the majority of the customers because it delivers unique hybrid capabilities to a customer, limited native storage options have restricted its usefulness for organizations with storage-heavy workloads. Because storage is directly tied to hosts, the only way to scale storage is to add more hosts, which can increase costs by 35-40% or more for storage intensive workloads. These workloads need additional storage and segregated performance, not additional horsepower, but that
means paying for additional hosts. This is where the recent integration of FSx for ONTAP comes in handy for storage and performance intensive workloads with VMware Cloud on AWS.

Let's consider the following scenario: a customer requires eight hosts for horsepower (vCPU/vMem), but they also have a substantial requirement for storage. Based on their assessment, they require 16 hosts to meet storage requirements. This increases the overall TCO because they must buy all that additional horsepower when all they really need is more storage. This is applicable for any use case, including migration, disaster recovery, bursting, dev/test, and so on.

This document walks you through the steps necessary to provision and attach FSx for ONTAP as a NFS datastore for VMware Cloud on AWS.

This solution is also available from VMware. Please visit the VMware Cloud Tech Zone for more information.

Connectivity options

VMware Cloud on AWS supports both multi-AZ and single-AZ deployments of FSx for ONTAP.

This section describes the high-level connectivity architecture along with the steps needed to implement the solution to expand the storage in a SDDC cluster without the need for adding additional hosts.

The high-level deployment steps are as follows:

1. Create Amazon FSx for ONTAP in a new designated VPC.
2. Create an SDDC group.
3. Create VMware Transit Connect and a TGW attachment.
4. Configure routing (AWS VPC and SDDC) and security groups.
5. Attach an NFS volume as a datastore to the SDDC cluster.

Before you provision and attach FSx for ONTAP as a NFS datastore, you must first set up a VMware on Cloud SDDC environment or get an existing SDDC upgraded to v1.20 or above. For more information, see the Getting Started With VMware Cloud on AWS.

FSx for ONTAP is not currently supported with stretched clusters.

Conclusion

This document covers the steps necessary to configure Amazon FSx for ONTAP with VMware cloud on AWS. Amazon FSx for ONTAP provides excellent options to deploy and manage application workloads along with file services while reducing the TCO by making data requirements seamless to the application layer. Whatever the use case, choose VMware Cloud on AWS along with Amazon FSx for ONTAP for rapid realization of cloud benefits, consistent infrastructure, and operations from on-premises to AWS, bidirectional portability of workloads, and enterprise-grade capacity and performance. It is the same familiar process and procedures used to connect storage. Remember, it is just the position of the data that changed along with new names; the tools and processes all remain the same, and Amazon FSx for ONTAP helps to optimize the overall deployment.

To learn more about this process, feel free to follow the detailed walkthrough video.

Amazon FSX for Ontap VMware Cloud

NetApp Guest Connected Storage Options for AWS

AWS supports guest connected NetApp storage with the native FSx service (FSx ONTAP) or with Cloud Volumes ONTAP (CVO).

FSx ONTAP

Amazon FSx for NetApp ONTAP is a fully managed service that provides highly reliable, scalable, high-performing, and feature-rich file storage built on NetApp’s popular ONTAP file system. FSx for ONTAP combines the familiar features, performance, capabilities, and API operations of NetApp file systems with the agility, scalability, and simplicity of a fully managed AWS service.

FSx for ONTAP provides feature-rich, fast, and flexible shared file storage that’s broadly accessible from Linux, Windows, and macOS compute instances running in AWS or on premises. FSx for ONTAP offers high-performance solid state drive (SSD) storage with submillisecond latencies. With FSx for ONTAP, you can achieve SSD levels of performance for your workload while paying for SSD storage for only a small fraction of your data.

Managing your data with FSx for ONTAP is easier because you can snapshot, clone, and replicate your files with the click of a button. In addition, FSx for ONTAP automatically tiers your data to lower-cost, elastic storage, lessening the need for you to provision or manage capacity.

FSx for ONTAP also provides highly available and durable storage with fully managed backups and support for cross-Region disaster recovery. To make it easier to protect and secure your data, FSx for ONTAP supports popular data security and antivirus applications.

FSx ONTAP as guest connected storage
Amazon FSx for NetApp ONTAP files shares and LUNs can be mounted from VMs that are created within the VMware SDDC environment at VMware Cloud at AWS. The volumes can also be mounted on the Linux client and mapped on the Windows client using the NFS or SMB protocol, and LUNS can be accessed on Linux or Windows clients as block devices when mounted over iSCSI. Amazon FSx for the NetApp ONTAP file system can be set up quickly with the following steps.

Amazon FSx for NetApp ONTAP and VMware Cloud on AWS must be in the same availability zone to achieve better performance and avoid data transfer charges between availability zones.
Create and mount Amazon FSx for ONTAP volumes

To create and mount Amazon FSx for NetApp ONTAP file system, complete the following steps:

1. Open the Amazon FSx console and choose Create file system to start the file system creation wizard.
2. On the Select File System Type page, choose Amazon FSx for NetApp ONTAP, and then choose Next. The Create File System page appears.

1. In the Networking section, for Virtual Private Cloud (VPC), choose the appropriate VPC and preferred subnets along with the route table. In this case, vmcfsx2.vpc is selected from the dropdown.

Create file system

Creation method

- Quick create
  Use recommended best-practice configurations. Most configuration options can be changed after the file system is created.
- Standard create
  You set all of the configuration options, including specifying performance, networking, security, backups, and maintenance.

1. For the creation method, choose Standard Create. You can also choose Quick Create, but this document uses the Standard create option.
1. In the Networking section, for Virtual Private Cloud (VPC), choose the appropriate VPC and preferred subnets along with the route table. In this case, vmcfsx2.vpc is selected from the dropdown.

<table>
<thead>
<tr>
<th>Network &amp; security</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Virtual Private Cloud (VPC)</strong></td>
</tr>
<tr>
<td>Specify the VPC from which your file system is accessible.</td>
</tr>
<tr>
<td>vmcfsx2.vpc</td>
</tr>
<tr>
<td><strong>VPC Security Groups</strong></td>
</tr>
<tr>
<td>Specify VPC Security Groups to associate with your file system’s network interface.</td>
</tr>
<tr>
<td>Choose VPC security group(s)</td>
</tr>
<tr>
<td>sg-01889a218164c80 (default)</td>
</tr>
<tr>
<td>Preferred subnet</td>
</tr>
<tr>
<td>Specify the preferred subnet for your file system.</td>
</tr>
<tr>
<td>subnet02 sn</td>
</tr>
<tr>
<td>Standby subnet</td>
</tr>
<tr>
<td>subnet01 sn</td>
</tr>
<tr>
<td><strong>VPC route tables</strong></td>
</tr>
<tr>
<td>Specify the VPC route tables associated with your file system.</td>
</tr>
<tr>
<td>○ VPC’s default route table</td>
</tr>
<tr>
<td>○ Select one or more VPC route tables</td>
</tr>
<tr>
<td><strong>Endpoint IP address range</strong></td>
</tr>
<tr>
<td>Specify the IP address range in which the endpoints to access your file system will be created</td>
</tr>
<tr>
<td>○ No preference</td>
</tr>
<tr>
<td>○ Select an IP address range</td>
</tr>
</tbody>
</table>
In the Networking section, for Virtual Private Cloud (VPC), choose the appropriate VPC and preferred subnets along with the route table. In this case, vmcfsx2.vpc is selected from the dropdown.

1. In the Security & Encryption section, for the Encryption Key, choose the AWS Key Management Service (AWS KMS) encryption key that protects the file system’s data at rest. For the File System Administrative Password, enter a secure password for the fsxadmin user.

Security & encryption

Encryption key

AWS Key Management Service (KMS) encryption key that protects your file system data at rest.

<table>
<thead>
<tr>
<th>Description</th>
<th>Account</th>
<th>KMS key ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default master key that protects my FSx resources when no other key is defined</td>
<td>139763910815</td>
<td>72745367-7bb0-499c-acc0-4f2c0a80e7c5</td>
</tr>
</tbody>
</table>

File system administrative password

Password for this file system's "fsxadmin" user, which you can use to access the ONTAP CLI or REST API.

- Don’t specify a password
- Specify a password

Password

Confirm password

1. In virtual machine and specify the password to use with vsadmin for administering ONTAP using REST APIs or the CLI. If no password is specified, a fsxadmin user can be used for administering the SVM. In the Active Directory section, make sure to join Active Directory to the SVM for provisioning SMB shares. In the Default Storage Virtual Machine Configuration section, provide a name for the storage in this validation, SMB shares are provisioned using a self-managed Active Directory domain.
1. In the Default Volume Configuration section, specify the volume name and size. This is an NFS volume. For Storage Efficiency, choose Enabled to turn on the ONTAP storage efficiency features (compression, deduplication, and compaction) or Disabled to turn them off.

<table>
<thead>
<tr>
<th>Storage virtual machine name</th>
<th>vmcfsvmval2svm</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM administrative password</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td></td>
</tr>
<tr>
<td>Confirm password</td>
<td></td>
</tr>
<tr>
<td>Active Directory</td>
<td></td>
</tr>
<tr>
<td>Do not join an Active Directory</td>
<td></td>
</tr>
<tr>
<td>Join an Active Directory</td>
<td></td>
</tr>
</tbody>
</table>

### Default volume configuration

<table>
<thead>
<tr>
<th>Volume name</th>
<th>vol1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum of 203 alphanumeric characters, plus _ .</td>
<td></td>
</tr>
<tr>
<td>Junction path</td>
<td>/vol1</td>
</tr>
<tr>
<td>The location within your file system where your volume will be mounted.</td>
<td></td>
</tr>
<tr>
<td>Volume size</td>
<td>1024</td>
</tr>
<tr>
<td>Minimum 20 MiB; Maximum 104857600 MiB</td>
<td></td>
</tr>
</tbody>
</table>

### Storage efficiency

Select whether you would like to enable ONTAP storage efficiencies on your volume: deduplication, compression, and compaction.

- Enabled (recommended)
- Disabled

### Capacity pool tiering policy

You can optionally enable automatic tiering of your data to lower-cost capacity pool storage.

- Auto
1. Review the file system configuration shown on the Create File System page.
2. Click Create File System.

For more detailed information, see Getting started with Amazon FSx for NetApp ONTAP.
After the file system is created as above, create the volume with the required size and protocol.

1. Open the Amazon FSx console.
2. In the left navigation pane, choose File systems, and then choose the ONTAP file system that you want to create a volume for.
3. Select the Volumes tab.
4. Select the Create Volume tab.
5. The Create Volume dialog box appears.

For demo purposes, an NFS volume is created in this section that can be easily mounted on VMs running on VMware cloud on AWS. nfsdemosvol01 is created as depicted below:
Mount FSx ONTAP volume on Linux client

To mount the FSx ONTAP volume created in the previous step, from the Linux VMs within VMC on AWS SDDC, complete the following steps:

1. Connect to the designated Linux instance.
2. Open a terminal on the instance using Secure Shell (SSH) and log in with the appropriate credentials.
3. Make a directory for the volume’s mount point with the following command:

   ```
   $ sudo mkdir /fsx/nfsdemovol01
   ```

4. Mount the Amazon FSx for NetApp ONTAP NFS volume to the directory that is created in the previous step.

   ```
   sudo mount -t nfs nfsvers=4.1,198.19.254.239:/nfsdemovol01
   /fsx/nfsdemovol01
   ```

   1. Once executed, run the `df` command to validate the mount.
Attach FSx ONTAP volumes to Microsoft Windows clients

To manage and map file shares on an Amazon FSx file system, the Shared Folders GUI must be used.

1. Open the Start menu and run fsmgmt.msc using Run As Administrator. Doing this opens the Shared Folders GUI tool.
2. Click Action > All tasks and choose Connect to Another Computer.
3. For Another Computer, enter the DNS name for the storage virtual machine (SVM). For example, FSXSMBTESTING01.FSXTESTING.LOCAL is used in this example.

To find the SVM's DNS name on the Amazon FSx console, choose Storage Virtual Machines, choose SVM, and then scroll down to Endpoints to find the SMB DNS name. Click OK. The Amazon FSx file system appears in the list for the Shared Folders.

**Endpoints**

<table>
<thead>
<tr>
<th>Management DNS name</th>
</tr>
</thead>
<tbody>
<tr>
<td>svm-075dcfbe2cfa2ece9.fs-040eacc5d0ac31017.fsx-west-2.amazonaws.com</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Management IP address</th>
</tr>
</thead>
<tbody>
<tr>
<td>198.19.254.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NFS DNS name</th>
</tr>
</thead>
<tbody>
<tr>
<td>svm-075dcfbe2cfa2ece9.fs-040eacc5d0ac31017.fsx-west-2.amazonaws.com</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NFS IP address</th>
</tr>
</thead>
<tbody>
<tr>
<td>198.19.254.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SMB DNS name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSXSMBTESTING01.FSXTESTING.LOCAL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SMB IP address</th>
</tr>
</thead>
<tbody>
<tr>
<td>198.19.254.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>iSCSI DNS name</th>
</tr>
</thead>
<tbody>
<tr>
<td>iscsi.svm-075dcfbe2cfa2ece9.fs-040eacc5d0ac31017.fsx-west-2.amazonaws.com</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>iSCSI IP addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.222.2.224, 10.222.1.94</td>
</tr>
</tbody>
</table>

1. In the Shared Folders tool, choose Shares in the left pane to see the active shares for the Amazon FSx file system.
1. Now choose a new share and complete the Create a Shared Folder wizard.
To learn more about creating and managing SMB shares on an Amazon FSx file system, see Creating SMB Shares.

1. After connectivity is in place, the SMB share can be attached and used for application data. To accomplish this, Copy the share path and use the Map Network Drive option to mount the volume on the VM running on VMware Cloud on the AWS SDDC.
Connect a FSx for NetApp ONTAP LUN to a host using iSCSI

iSCSI traffic for FSx traverses the VMware Transit Connect/AWS Transit Gateway via the routes provided in the previous section. To configure a LUN in Amazon FSx for NetApp ONTAP, follow the documentation found here.

On Linux clients, make sure that the iSCSI daemon is running. After the LUNs are provisioned, refer to the detailed guidance on iSCSI configuration with Ubuntu (as an example) here.

In this paper, connecting the iSCSI LUN to a Windows host is depicted:
Provision a LUN in FSx for NetApp ONTAP:

1. Access the NetApp ONTAP CLI using the management port of the FSx for the ONTAP file system.
2. Create the LUNs with the required size as indicated by the sizing output.

   ```
   FsxId040eacc5d0ac31017::> lun create -vserver vmcfsxval2svm -volume nimfsxscsivol -lun nimofsxlun01 -size 5gb -ostype windows -space -reserve enabled
   ```

   In this example, we created a LUN of size 5g (5368709120).

1. Create the necessary igroups to control which hosts have access to specific LUNs.

   ```
   FsxId040eacc5d0ac31017::> igroup create -vserver vmcfsxval2svm -igroup winIG -protocol iscsi -ostype windows -initiator iqn.1991-05.com.microsoft:vmcdc01.fsxtesting.local
   ```

   ```
   FsxId040eacc5d0ac31017::> igroup show
   Vserver   Igroup       Protocol OS Type  Initiators
            --------- ------------ -------- --------
            ------------------------------------
            vmcfsxval2svm
            ubuntu01   iscsi   linux   iqn.2021-10.com.ubuntu:01:initiator01
            vmcfsxval2svm
            winIG       iscsi   windows iqn.1991-05.com.microsoft:vmcdc01.fsxtesting.local
   ```

   Two entries were displayed.

1. Map the LUNs to igroups using the following command:
Two entries were displayed.

1. Connect the newly provisioned LUN to a Windows VM:

To connect the new LUN to a Windows host residing on VMware cloud on AWS SDDC, complete the following steps:

a. RDP to the Windows VM hosted on the VMware Cloud on AWS SDDC.

b. Navigate to Server Manager > Dashboard > Tools > iSCSI Initiator to open the iSCSI Initiator Properties dialog box.

c. From the Discovery tab, click Discover Portal or Add Portal and then enter the IP address of the iSCSI target port.

d. From the Targets tab, select the target discovered and then click Log On or Connect.

e. Select Enable Multipath, and then select “Automatically Restore This Connection When the Computer Starts” or “Add This Connection to the List of Favorite Targets”. Click Advanced.

The Windows host must have an iSCSI connection to each node in the cluster. The native DSM selects the best paths to use.
LUNs on the storage virtual machine (SVM) appear as disks to the Windows host. Any new disks that are added are not automatically discovered by the host. Trigger a manual rescan to discover the disks by completing the following steps:

2. Expand the Storage node in the navigation tree.
3. Click Disk Management.
4. Click Action > Rescan Disks.
When a new LUN is first accessed by the Windows host, it has no partition or file system. Initialize the LUN and, optionally, format the LUN with a file system by completing the following steps:

2. Right-click the LUN, and then select the required disk or partition type.
3. Follow the instructions in the wizard. In this example, drive F: is mounted.

Cloud Volumes ONTAP (CVO)

Cloud volumes ONTAP, or CVO, is the industry-leading cloud data management solution built on NetApp’s ONTAP storage software, available natively on Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform (GCP).

It is a software-defined version of ONTAP that consumes cloud-native storage, allowing you to have the same storage software in the cloud and on-premises, reducing the need to retrain your IT staff in all-new methods to
manage your data.

CVO gives customers the ability to seamlessly move data from the edge, to the data center, to the cloud and back, bringing your hybrid cloud together — all managed with a single-pane management console, NetApp Cloud Manager.

By design, CVO delivers extreme performance and advanced data management capabilities to satisfy even your most demanding applications in the cloud

Cloud Volumes ONTAP (CVO) as guest connected storage
Deploy new Cloud Volumes ONTAP instance in AWS (do it yourself)

Cloud Volumes ONTAP shares and LUNs can be mounted from VMs that are created in the VMware Cloud on AWS SDDC environment. The volumes can also be mounted on native AWS VM Linux Windows clients, and LUNs can be accessed on Linux or Windows clients as block devices when mounted over iSCSI because Cloud Volumes ONTAP supports iSCSI, SMB, and NFS protocols. Cloud Volumes ONTAP volumes can be set up in a few simple steps.

To replicate volumes from an on-premises environment to the cloud for disaster recovery or migration purposes, establish network connectivity to AWS, either using a site-to-site VPN or DirectConnect. Replicating data from on-premises to Cloud Volumes ONTAP is outside the scope of this document. To replicate data between on-premises and Cloud Volumes ONTAP systems, see Setting up data replication between systems.

Use the Cloud Volumes ONTAP sizer to accurately size the Cloud Volumes ONTAP instances. Also, monitor on-premises performance to use as inputs in the Cloud Volumes ONTAP sizer.

1. Log into NetApp Cloud Central; the Fabric View screen is displayed. Locate the Cloud Volumes ONTAP tab and select Go to Cloud Manager. After you are logged in, the Canvas screen is displayed.

1. On the Cloud Manager home page, click Add a Working Environment and then select AWS as the cloud and the type of the system configuration.

1. Provide the details of the environment to be created including the environment name and admin credentials. Click Continue.
1. Select the add-on services for Cloud Volumes ONTAP deployment, including BlueXP Classification, BlueXP backup and recovery, and Cloud Insights. Click Continue.

1. On the HA Deployment Models page, choose the Multiple Availability Zones configuration.

1. On the Region & VPC page, enter the network information and then click Continue.
1. On the Connectivity and SSH Authentication page, choose connection methods for the HA pair and the mediator.

1. Specify the floating IP addresses and then click Continue.
1. Select the appropriate route tables to include routes to the floating IP addresses and then click Continue.

1. On the Data Encryption page, choose AWS-managed encryption.
1. Select the license option: Pay-As-You-Go or BYOL for using an existing license. In this example, the Pay-As-You-Go option is used.

Create a New Working Environment

Cloud Volumes ONTAP Charging Methods & NSS Account

Cloud Volumes ONTAP Charging Methods

Learn more about our charging methods

- Pay-As-You-Go by the hour
- Bring your own license

NetApp Support Site Account (Optional)

Learn more about NetApp Support Site (NSS) accounts

To register this Cloud Volumes ONTAP to support, you should add NetApp Support Site Account.

Don't have a NetApp Support Site account? Select go to finish deploying this system. After its created, use the Support Registration online to create an NSS account.

Continue

1. Select between several preconfigured packages available based on the type of workload to be deployed on the VMs running on the VMware cloud on AWS SDDC.

Create a New Working Environment

Preconfigured Packages

Select a preconfigured Cloud Volumes ONTAP system that best matches your needs, or create your own configuration. Preconfigured settings can be modified at a later time.

- POC and small workloads
  Up to 500GB of storage
- Database and application data production workloads
- Cost effective DR
  Up to 500GB of storage
- Highest performance production workloads

Continue
1. On the Review & Approve page, review and confirm the selections. To create the Cloud Volumes ONTAP instance, click Go.

1. After Cloud Volumes ONTAP is provisioned, it is listed in the working environments on the Canvas page.
Additional configurations for SMB volumes

1. After the working environment is ready, make sure the CIFS server is configured with the appropriate DNS and Active Directory configuration parameters. This step is required before you can create the SMB volume.

   ```
   fsxvcotesting01 (Multiple AZs)
   
   Create a CIFS server
   
   DNS Primary IP Address:
   192.168.1.1
   
   DNS Secondary IP Address (Optional):
   Example: 127.0.0.1
   
   ACTIVE DIRECTORY Domain to Join:
   fsxvcotesting.local
   
   Credentials authorized to join the domain:
   Username
   Password
   
   Save  Cancel
   ```

1. Select the CVO instance to create the volume and click the Create Volume option. Choose the appropriate size and cloud manager chooses the containing aggregate or use advanced allocation mechanism to place on a specific aggregate. For this demo, SMB is selected as the protocol.

   ```
   Create new volume in fsxvcotesting01
   
   Volume Details, Protection & Protocol
   
   Details & Protection
   
   Volume Name: smbdemovol01
   Size (GB): 10
   
   Snapshot Policy:
   default
   Default Policy
   
   Protocol
   
   NFS  CIFS  iSCSI
   
   Share name: smbdemovol01_share
   Permissions: Full Control
   
   Users / Groups:
   Everyone
   Valid users and groups separated by a semicolon
   
   Continue
   ```

1. After the volume is provisioned, it is available under the Volumes pane. Because a CIFS share is provisioned, you should give your users or groups permission to the files and folders and verify that those users can access the share and create a file.
1. After the volume is created, use the mount command to connect to the share from the VM running on the VMware Cloud in AWS SDDC hosts.

2. Copy the following path and use the Map Network Drive option to mount the volume on the VM running on the VMware Cloud in AWS SDDC.
Connect the LUN to a host

To connect the Cloud Volumes ONTAP LUN to a host, complete the following steps:

1. On the Cloud Manager Canvas page, double-click the Cloud Volumes ONTAP working environment to create and manage volumes.
2. Click Add Volume > New Volume, select iSCSI, and click Create Initiator Group. Click Continue.

1. After the volume is provisioned, select the volume, and then click Target IQN. To copy the iSCSI Qualified Name (IQN), click Copy. Set up an iSCSI connection from the host to the LUN.

To accomplish the same for the host residing on the VMware Cloud on AWS SDDC, complete the following steps:

a. RDP to the VM hosted on VMware cloud on AWS.
b. Open the iSCSI Initiator Properties dialog box: Server Manager > Dashboard > Tools > iSCSI Initiator.
c. From the Discovery tab, click Discover Portal or Add Portal and then enter the IP address of the iSCSI target port.
d. From the Targets tab, select the target discovered and then click Log On or Connect.
e. Select Enable Multipath, and then select Automatically Restore This Connection When the Computer Starts or Add This Connection to the List of Favorite Targets. Click Advanced.

The Windows host must have an iSCSI connection to each node in the cluster. The native DSM selects the best paths to use.

LUNs from the SVM appear as disks to the Windows host. Any new disks that are added are not automatically discovered by the host. Trigger a manual rescan to discover the disks by completing the following steps:

2. Expand the Storage node in the navigation tree.
3. Click Disk Management.
4. Click Action > Rescan Disks.
When a new LUN is first accessed by the Windows host, it has no partition or file system. Initialize the LUN; and optionally, format the LUN with a file system by completing the following steps:

2. Right-click the LUN, and then select the required disk or partition type.
3. Follow the instructions in the wizard. In this example, drive F: is mounted.

On the Linux clients, ensure the iSCSI daemon is running. After the LUNs are provisioned, refer to the detailed guidance on iSCSI configuration for your Linux distribution. For example, Ubuntu iSCSI configuration can be found here. To verify, run lsblk cmd from the shell.
Mount Cloud Volumes ONTAP NFS volume on Linux client

To mount the Cloud Volumes ONTAP (DIY) file system from VMs within VMC on AWS SDDC, complete the following steps:

1. Connect to the designated Linux instance.
2. Open a terminal on the instance using secure shell (SSH) and log in with the appropriate credentials.
3. Make a directory for the volume’s mount point with the following command.

```
$ sudo mkdir /fsxcvotesting01/nfsdemovol01
```

4. Mount the Amazon FSx for NetApp ONTAP NFS volume to the directory that is created in the previous step.

```
sudo mount -t nfs nfsvers=4.1,172.16.0.2:/nfsdemovol01
/fsxcvotesting01/nfsdemovol01
```

Overview of ANF Datastore Solutions

Every successful organization is on a path of transformation and modernization. As part of this process, companies typically use their existing VMware investments while leveraging cloud benefits and exploring how to make migration, burst, extend, and disaster recovery processes as seamless as possible. Customers migrating to the cloud must evaluate the issues of elasticity and burst, data center exit, data center consolidation, end-of-life scenarios, mergers, acquisitions, and so on. The approach adopted by each organization can vary based on their respective business priorities. When choosing cloud-based operations, selecting a low-cost model with appropriate performance and minimal hindrance is a critical goal. Along with choosing the right platform, storage and workflow orchestration is particularly important to unleash the power of cloud deployment and elasticity.
Use Cases

Although the Azure VMware solution delivers unique hybrid capabilities to a customer, limited native storage options have restricted its usefulness for organizations with storage-heavy workloads. Because storage is directly tied to hosts, the only way to scale storage is to add more hosts, which can increase costs by 35-40% or more for storage intensive workloads. These workloads need additional storage, not additional horsepower, but that means paying for additional hosts.

Let’s consider the following scenario; a customer requires six hosts for horsepower (vCPU/vMem), but they also have a substantial requirement for storage. Based on their assessment, they require 12 hosts to meet storage requirements. This increases the overall TCO because they must buy all that additional horsepower when all they really need is more storage. This is applicable for any use case, including migration, disaster recovery, bursting, dev/test, and so on.

Another common use case for Azure VMware Solution is disaster recovery (DR). Most organizations do not have a fool-proof DR strategy, or they might struggle to justify running a ghost datacenter just for DR. Administrators might explore zero-footprint DR options with a pilot-light cluster or an on-demand cluster. They could then scale the storage without adding additional hosts, potentially an attractive option.

So, to summarize, the use cases can be classified in two ways:

- Scaling storage capacity using ANF datastores
- Using ANF datastores as a disaster recovery target for a cost-optimized recovery workflow from on-premises or within Azure regions between the software-defined datacenters (SDDCs). This guide provides insight into using Azure NetApp Files to provide optimized storage for datastores (currently in public preview) along with best-in-class data protection and DR capabilities in an Azure VMware solution, which enables you to offload storage capacity from vSAN storage.

Contact NetApp or Microsoft solution architects in your region for additional information on using ANF datastores.

VMware Cloud options in Azure

Azure VMware Solution

The Azure VMware Solution (AVS) is a hybrid cloud service that provides fully functioning VMware SDDCs within a Microsoft Azure public cloud. AVS is a first-party solution fully managed and supported by Microsoft and verified by VMware that uses Azure infrastructure. Therefore, customers get VMware ESXi for compute virtualization, vSAN for hyper-converged storage, and NSX for networking and security, all while taking advantage of Microsoft Azure’s global presence, class-leading data center facilities, and proximity to the rich ecosystem of native Azure services and solutions. A combination of Azure VMware Solution SDDC and Azure NetApp Files provides the best performance with minimal network latency.

Regardless of the cloud used, when a VMware SDDC is deployed, the initial cluster includes the following components:

- VMware ESXi hosts for compute virtualization with a vCenter server appliance for management.
- VMware vSAN hyper-converged storage incorporating the physical storage assets of each ESXi host.
- VMware NSX for virtual networking and security with an NSX Manager cluster for management.

Conclusion

Whether you are targeting all-cloud or hybrid cloud, Azure NetApp files provide excellent options to deploy and
manage the application workloads along with file services while reducing the TCO by making the data requirements seamless to the application layer. Whatever the use case, choose Azure VMware Solution along with Azure NetApp Files for rapid realization of cloud benefits, consistent infrastructure, and operations across on-premises and multiple clouds, bi-directional portability of workloads, and enterprise-grade capacity and performance. It is the same familiar process and procedures used to connect the storage. Remember, it is just the position of the data that changed along with new names; the tools and processes all remain the same, and Azure NetApp Files helps in optimizing the overall deployment.

Takeaways

The key points of this document include:

- You can now use Azure NetApp Files as a datastore on AVS SDDC.
- Boost the application response times and deliver higher availability to provide access workload data when and where it is needed.
- Simplify the overall complexity of the vSAN storage with simple and instant resizing capabilities.
- Guaranteed performance for mission-critical workloads using dynamic reshaping capabilities.
- If Azure VMware Solution Cloud is the destination, Azure NetApp Files is the right storage solution for optimized deployment.

Where to find additional information

To learn more about the information described in this document, refer to the following website links:

- Azure VMware Solution documentation
  https://docs.microsoft.com/en-us/azure/azure-vmware/
- Azure NetApp Files documentation
  https://docs.microsoft.com/en-us/azure/azure-netapp-files/
- Attach Azure NetApp Files datastores to Azure VMware Solution hosts (Preview)

NetApp Guest Connected Storage Options for Azure

Azure supports guest connected NetApp storage with the native Azure NetApp Files (ANF) service or with Cloud Volumes ONTAP (CVO).

Azure NetApp Files (ANF)

Azure NetApp Files brings enterprise-grade data management and storage to Azure so you can manage your workloads and applications with ease. Migrate your workloads to the cloud and run them without sacrificing performance.

Azure NetApp Files removes obstacles, so you can move all of your file-based applications to the cloud. For the first time, you do not have to re-architect your applications, and you get persistent storage for your applications without complexity.

Because the service is delivered through the Microsoft Azure Portal, users experience a fully managed service
as part of their Microsoft enterprise Agreement. World-class support, managed by Microsoft, gives you complete peace of mind. This single solution enables you to quickly and easily add multiprotocol workloads. you can build and deploy both Windows and Linux file-based applications, even for legacy environments.

**Azure NetApp Files (ANF) as guest connected storage**

**Configure Azure NetApp Files with Azure VMware Solution (AVS)**

Azure NetApp Files shares can be mounted from VMs that are created in the Azure VMware Solution SDDC environment. The volumes can also be mounted on the Linux client and mapped on the Windows client because Azure NetApp Files supports SMB and NFS protocols. Azure NetApp Files volumes can be set up in five simple steps.

Azure NetApp Files and Azure VMware Solution must be in the same Azure region.
Create and mount Azure NetApp Files volumes

To create and mount Azure NetApp Files volumes, complete the following steps:


   For detailed steps, see Azure NetApp Files shares. This page will guide you through the step-by-step process.

2. After the NetApp account is created, set up the capacity pools with the required service level and size.

   For more information, see Set up a capacity pool.
3. Configure the delegated subnet for Azure NetApp Files and specify this subnet while creating the volumes. For detailed steps to create delegated subnet, see Delegate a subnet to Azure NetApp Files.

4. Add an SMB volume by using the Volumes blade under the Capacity Pools blade. Make sure the Active Directory connector is configured prior to creating the SMB volume.
5. Click Review + Create to create the SMB volume.

If the application is SQL Server, then enable the SMB continuous availability.
To learn more about Azure NetApp Files volume performance by size or quota, see Performance considerations for Azure NetApp Files.

6. After the connectivity is in place, the volume can be mounted and used for application data.

To accomplish this, from the Azure portal, click the Volumes blade, and then select the volume to mount and access the mount instructions. Copy the path and use the Map Network Drive option to mount the volume on the VM running on Azure VMware Solution SDDC.
7. To mount NFS volumes on Linux VMs running on Azure VMware Solution SDDC, use this same process. Use volume reshaping or dynamic service level capability to meet the workload demands.

For more information, see [Dynamically change the service level of a volume](#).

### Cloud Volumes ONTAP (CVO)

Cloud volumes ONTAP, or CVO, is the industry-leading cloud data management solution built on NetApp’s ONTAP storage software, available natively on Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform (GCP).

It is a software-defined version of ONTAP that consumes cloud-native storage, allowing you to have the same storage software in the cloud and on-premises, reducing the need to retrain your IT staff in all-new methods to manage your data.
CVO gives customers the ability to seamlessly move data from the edge, to the data center, to the cloud and back, bringing your hybrid cloud together — all managed with a single-pane management console, NetApp Cloud Manager.

By design, CVO delivers extreme performance and advanced data management capabilities to satisfy even your most demanding applications in the cloud

**Cloud Volumes ONTAP (CVO) as guest connected storage**
Deploy new Cloud Volumes ONTAP in Azure

Cloud Volumes ONTAP shares and LUNs can be mounted from VMs that are created in the Azure VMware Solution SDDC environment. The volumes can also be mounted on the Linux client and on Windows client because Cloud Volumes ONTAP supports iSCSI, SMB, and NFS protocols. Cloud Volumes ONTAP volumes can be set up in a few simple steps.

To replicate volumes from an on-premises environment to the cloud for disaster recovery or migration purposes, establish network connectivity to Azure, either using a site-to-site VPN or ExpressRoute. Replicating data from on-premises to Cloud Volumes ONTAP is outside the scope of this document. To replicate data between on-premises and Cloud Volumes ONTAP systems, see Setting up data replication between systems.

Use Cloud Volumes ONTAP sizer to accurately size the Cloud Volumes ONTAP instances. Also monitor on-premises performance to use as inputs in the Cloud Volumes ONTAP sizer.

1. Log in to NetApp Cloud Central—the Fabric View screen is displayed. Locate the Cloud Volumes ONTAP tab and select Go to Cloud Manager. After you are logged in, the Canvas screen is displayed.

2. On the Cloud Manager home page, click Add a Working Environment and then select Microsoft Azure as the cloud and the type of the system configuration.
3. When creating the first Cloud Volumes ONTAP working environment, Cloud Manager prompts you to deploy a Connector.

4. After the connector is created, update the Details and Credentials fields.
5. Provide the details of the environment to be created including the environment name and admin credentials. Add resource group tags for the Azure environment as an optional parameter. After you are done, click Continue.

6. Select the add-on services for Cloud Volumes ONTAP deployment, including BlueXP Classification, BlueXP backup and recovery, and Cloud Insights. Select the services and then click Continue.

7. Configure the Azure location and connectivity. Select the Azure Region, resource group, VNet, and subnet to be used.
8. Select the license option: Pay-As-You-Go or BYOL for using existing license. In this example, Pay-As-You-Go option is used.

9. Select between several preconfigured packages available for the various types of workloads.

10. Accept the two agreements regarding activating support and allocation of Azure resources. To create the Cloud Volumes ONTAP instance, click Go.
11. After Cloud Volumes ONTAP is provisioned, it is listed in the working environments on the Canvas page.
Additional configurations for SMB volumes

1. After the working environment is ready, make sure the CIFS server is configured with the appropriate DNS and Active Directory configuration parameters. This step is required before you can create the SMB volume.

![Creating a CIFS server](image)

2. Creating the SMB volume is an easy process. Select the CVO instance to create the volume and click the Create Volume option. Choose the appropriate size and cloud manager chooses the containing aggregate or use advanced allocation mechanism to place on a specific aggregate. For this demo, SMB is selected as the protocol.

![Create new volume in nimavsCVO](image)

3. After the volume is provisioned, it will be available under the Volumes pane. Because a CIFS share is provisioned, give your users or groups permission to the files and folders and verify that those users can access the share and create a file. This step is not required if the volume is replicated from an on-premises environment because the file and folder permissions are all retained as part of SnapMirror replication.
4. After the volume is created, use the mount command to connect to the share from the VM running on the Azure VMware Solution SDDC hosts.

5. Copy the following path and use the Map Network Drive option to mount the volume on the VM running on Azure VMware Solution SDDC.

```
\172.24.2.8\nimavssmbvol1_share
```

Go to your machine and enter this command.
Connect the LUN to a host

To connect the LUN to a host, complete the following steps:

1. On the Canvas page, double-click the Cloud Volumes ONTAP working environment to create and manage volumes.
2. Click Add Volume > New Volume and select iSCSI and click Create Initiator Group. Click Continue.

![Details & Protection and Protocol](image)

3. After the volume is provisioned, select the volume, and then click Target IQN. To copy the iSCSI Qualified Name (IQN), click Copy. Set up an iSCSI connection from the host to the LUN.

To accomplish the same for the host residing on Azure VMware Solution SDDC:

a. RDP to the VM hosted on Azure VMware Solution SDDC.

b. Open the iSCSI Initiator Properties dialog box: Server Manager > Dashboard > Tools > iSCSI Initiator.

c. From the Discovery tab, click Discover Portal or Add Portal and then enter the IP address of the iSCSI target port.

d. From the Targets tab, select the target discovered and then click Log on or Connect.

e. Select Enable multipath, and then select Automatically Restore This Connection When the Computer Starts or Add This Connection to the List of Favorite Targets. Click Advanced.

**Note:** The Windows host must have an iSCSI connection to each node in the cluster. The native DSM selects the best paths to use.
LUNs on storage virtual machine (SVM) appear as disks to the Windows host. Any new disks that are added are not automatically discovered by the host. Trigger a manual rescan to discover the disks by completing the following steps:

2. Expand the Storage node in the navigation tree.
3. Click Disk Management.
4. Click Action > Rescan Disks.

When a new LUN is first accessed by the Windows host, it has no partition or file system. Initialize the LUN; and optionally, format the LUN with a file system by completing the following steps:
Google Cloud VMware Engine Supplemental NFS Datastore with NetApp Cloud Volume Service

Overview

Authors: Suresh Thoppay, NetApp

Customers that requires additional storage capacity on their Google Cloud VMware Engine (GCVE) environment can utilize Netapp Cloud Volume Service to mount as supplemental NFS datastore. Storing data on NetApp Cloud Volume Service allows customers to replicate between regions to protect from disaster.
Deployment steps to mount NFS datastore from NetApp CVS on GCVE

Provision CVS-Performance Volume

The NetApp Cloud Volume Service volume can be either provisioned by
Using Google Cloud Console
Using NetApp BlueXP portal or API
Mark that CVS volume as non-deletable

To avoid accidental deletion of volume while VM is running, ensure the volume is marked as non-deletable as shown in screenshot below.

Ensure Private Connection on GCVE exists for NetApp CVS Tenant VPC.

For more info, please refer Creating NFS Volume documentation.

For more info, please refer How to setup Private Service Access
Mount NFS datastore

For instructions on how to mount NFS datastore on GCVE, please refer [How to create NFS datastore with NetApp CVS](#).

As vSphere hosts are managed by Google, you don’t have access to install NFS vSphere API for Array Integration (VAAI) vSphere Installation Bundle (VIB). If you need support for Virtual Volumes (vVol), please let us know. If you like to use Jumbo Frames, please refer [Maximum supported MTU sizes on GCP](#).

Savings with NetApp Cloud Volume Service

To learn more about your potential saving with NetApp Cloud Volume Service for your storage demands on GCVE, please check [NetApp ROI Calculator](#).

Reference Links

- Google Blog - How to use NetApp CVS as datastores for Google Cloud VMware Engine
- NetApp Blog - A better way to migrate your storage-rich apps to Google Cloud

NetApp Storage Options for GCP

GCP supports guest connected NetApp storage with Cloud Volumes ONTAP (CVO) or Cloud Volumes Service (CVS).

Cloud Volumes ONTAP (CVO)

Cloud volumes ONTAP, or CVO, is the industry-leading cloud data management solution built on NetApp’s ONTAP storage software, available natively on Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform (GCP).

It is a software-defined version of ONTAP that consumes cloud-native storage, allowing you to have the same storage software in the cloud and on-premises, reducing the need to retrain you IT staff in all-new methods to manage your data.

CVO gives customers the ability to seamlessly move data from the edge, to the data center, to the cloud and back, bringing your hybrid cloud together — all managed with a single-pane management console, NetApp Cloud Manager.

By design, CVO delivers extreme performance and advanced data management capabilities to satisfy even your most demanding applications in the cloud.

Cloud Volumes ONTAP (CVO) as guest connected storage
Cloud Volumes ONTAP shares and LUNs can be mounted from VMs that are created in the GCVE private cloud environment. The volumes can also be mounted on the Linux client and on Windows client and LUNS can be accessed on Linux or Windows clients as block devices when mounted over iSCSI because Cloud Volumes ONTAP supports iSCSI, SMB, and NFS protocols. Cloud Volumes ONTAP volumes can be set up in a few simple steps.

To replicate volumes from an on-premises environment to the cloud for disaster recovery or migration purposes, establish network connectivity to Google Cloud, either using a site-to-site VPN or Cloud Interconnect. Replicating data from on-premises to Cloud Volumes ONTAP is outside the scope of this document. To replicate data between on-premises and Cloud Volumes ONTAP systems, see xref:./ehc/Setting up data replication between systems.

Use Cloud Volumes ONTAP sizer to accurately size the Cloud Volumes ONTAP instances. Also monitor on-premises performance to use as inputs in the Cloud Volumes ONTAP sizer.

1. Log in to NetApp Cloud Central—the Fabric View screen is displayed. Locate the Cloud Volumes ONTAP tab and select Go to Cloud Manager. After you are logged in, the Canvas screen is displayed.

2. On the Cloud Manager Canvas tab, click Add a Working Environment and then select Google Cloud Platform as the cloud and the type of the system configuration. Then, click Next.

3. Provide the details of the environment to be created including the environment name and admin
credentials. After you are done, click Continue.

4. Select or deselect the add-on services for Cloud Volumes ONTAP deployment, including Data Sense & Compliance or Backup to Cloud. Then, click Continue.

HINT: A verification pop-up message will be displayed when deactivating add-on services. Add-on services can be added/removed after CVO deployment, consider to deselect them if not needed from the beginning to avoid costs.

5. Select a location, choose a firewall policy, and select the checkbox to confirm network connectivity to Google Cloud storage.
6. Select the license option: Pay-As-You-Go or BYOL for using existing license. In this example, Freemium option is used. Then, click on Continue.

7. Select between several preconfigured packages available based on the type of workload that will be deployed on the VMs running on VMware cloud on AWS SDDC.

HINT: Hoover your mouse over the tiles for details or customize CVO components and ONTAP version by clicking on Change Configuration.
8. On the Review & Approve page, review and confirm the selections. To create the Cloud Volumes ONTAP instance, click Go.

9. After Cloud Volumes ONTAP is provisioned, it is listed in the working environments on the Canvas page.
Additional configurations for SMB volumes

1. After the working environment is ready, make sure the CIFS server is configured with the appropriate DNS and Active Directory configuration parameters. This step is required before you can create the SMB volume.

   HINT: Click on the Menu Icon (º), select Advanced to display more options and select CIFS setup.

2. Creating the SMB volume is an easy process. At Canvas, double-click the Cloud Volumes ONTAP working environment to create and manage volumes and click on the Create Volume option. Choose the appropriate size and cloud manager chooses the containing aggregate or use advanced allocation mechanism to place on a specific aggregate. For this demo, CIFS/SMB is selected as the protocol.

3. After the volume is provisioned, it will be available under the Volumes pane. Because a CIFS share is provisioned, give your users or groups permission to the files and folders and verify that those users can access the share and create a file. This step is not required if the volume is replicated from an on-premises environment because the file and folder permissions are all retained as part of SnapMirror replication.

   HINT: Click on the volume menu (º) to display its options.
4. After the volume is created, use the mount command to display the volume connection instructions, then connect to the share from the VMs on Google Cloud VMware Engine.

5. Copy the following path and use the Map Network Drive option to mount the volume on the VM running on the Google Cloud VMware Engine.

Specify the drive letter for the connection and the folder that you want to connect to:

- **Drive:**
  - **Y:**

- **Folder:**
  - \10.0.6.251\cvogcvesmbsvol01_share

Example: \server\share
- **Reconnect at sign-in**
- **Connect using different credentials**

Connect to a Web site that you can use to store your documents and pictures.
Once mapped, it can be easily accessed, and the NTFS permissions can be set accordingly.
Connect the LUN on Cloud Volumes ONTAP to a host

To connect the cloud volumes ONTAP LUN to a host, complete the following steps:

1. On the Canvas page, double-click the Cloud Volumes ONTAP working environment to create and manage volumes.
2. Click Add Volume > New Volume and select iSCSI and click Create Initiator Group. Click Continue.
3. After the volume is provisioned, select the volume menu (⁎), and then click Target iQN. To copy the iSCSI Qualified Name (iQN), click Copy. Set up an iSCSI connection from the host to the LUN.

To accomplish the same for the host residing on Google Cloud VMware Engine:

a. RDP to the VM hosted on Google Cloud VMware Engine.

b. Open the iSCSI Initiator Properties dialog box: Server Manager > Dashboard > Tools > iSCSI Initiator.

c. From the Discovery tab, click Discover Portal or Add Portal and then enter the IP address of the iSCSI target port.
d. From the Targets tab, select the target discovered and then click Log on or Connect.

e. Select Enable multipath, and then select Automatically Restore This Connection When the Computer Starts or Add This Connection to the List of Favorite Targets. Click Advanced.

The Windows host must have an iSCSI connection to each node in the cluster. The native DSM selects the best paths to use.

LUNs on storage virtual machine (SVM) appear as disks to the Windows host. Any new disks that are added are not automatically discovered by the host. Trigger a manual rescan to discover the disks by completing the following steps:

2. Expand the Storage node in the navigation tree.
3. Click Disk Management.
4. Click Action > Rescan Disks.
When a new LUN is first accessed by the Windows host, it has no partition or file system. Initialize the LUN; and optionally, format the LUN with a file system by completing the following steps:

5. Start Windows Disk Management.
6. Right-click the LUN, and then select the required disk or partition type.
7. Follow the instructions in the wizard. In this example, drive F: is mounted.
On the Linux clients, ensure the iSCSI daemon is running. Once the LUNs are provisioned, refer to the detailed guidance on iSCSI configuration with Ubuntu as an example here. To verify, run `lsblk` cmd from the shell.
<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Size</th>
<th>Used</th>
<th>Avail</th>
<th>Use%</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>udev</td>
<td>1.9G</td>
<td>0.9G</td>
<td>0.9G</td>
<td>0%</td>
<td>/dev</td>
</tr>
<tr>
<td>/dev/sda5</td>
<td>16G</td>
<td>7.6G</td>
<td>8.4G</td>
<td>53%</td>
<td></td>
</tr>
<tr>
<td>tmpfs</td>
<td>2.0G</td>
<td>2.0G</td>
<td>0.0G</td>
<td>0%</td>
<td>/dev/shm</td>
</tr>
<tr>
<td>tmpfs</td>
<td>5.0M</td>
<td>5.0M</td>
<td>0.0M</td>
<td>0%</td>
<td>/run/lock</td>
</tr>
<tr>
<td>tmpfs</td>
<td>2.0G</td>
<td>2.0G</td>
<td>0.0G</td>
<td>0%</td>
<td>/sys/fs/cgroup</td>
</tr>
<tr>
<td>/dev/loop1</td>
<td>219M</td>
<td>199M</td>
<td>20M</td>
<td>90%</td>
<td>/snap/gnome-3.34-1804/72</td>
</tr>
<tr>
<td>/dev/loop2</td>
<td>66M</td>
<td>66M</td>
<td>0.0M</td>
<td>100%</td>
<td>/snap/gtk-common-themes/1515</td>
</tr>
<tr>
<td>/dev/loop3</td>
<td>51M</td>
<td>51M</td>
<td>0.0M</td>
<td>100%</td>
<td>/snap/snap-store/547</td>
</tr>
<tr>
<td>/dev/loop0</td>
<td>56M</td>
<td>56M</td>
<td>0.0M</td>
<td>100%</td>
<td>/snap/core18/2120</td>
</tr>
<tr>
<td>/dev/loop4</td>
<td>33M</td>
<td>33M</td>
<td>0.0M</td>
<td>100%</td>
<td>/snap/snapd/12704</td>
</tr>
<tr>
<td>/dev/sda1</td>
<td>511M</td>
<td>491M</td>
<td>20M</td>
<td>9%</td>
<td>/boot/efi</td>
</tr>
<tr>
<td>tmpfs</td>
<td>394M</td>
<td>394M</td>
<td>0.0M</td>
<td>100%</td>
<td>/run/user/1000</td>
</tr>
<tr>
<td>/dev/loop5</td>
<td>33M</td>
<td>33M</td>
<td>0.0M</td>
<td>100%</td>
<td>/snap/snapd/13640</td>
</tr>
<tr>
<td>/dev/loop6</td>
<td>56M</td>
<td>56M</td>
<td>0.0M</td>
<td>100%</td>
<td>/snap/core18/2246</td>
</tr>
<tr>
<td>/dev/loop7</td>
<td>128K</td>
<td>128K</td>
<td>0.0M</td>
<td>100%</td>
<td>/snap/bare/s</td>
</tr>
<tr>
<td>/dev/loop8</td>
<td>66M</td>
<td>66M</td>
<td>0.0M</td>
<td>100%</td>
<td>/snap/gtk-common-themes/1519</td>
</tr>
<tr>
<td>/dev/sdb</td>
<td>976M</td>
<td>2.6M</td>
<td>907M</td>
<td>1%</td>
<td>/mnt</td>
</tr>
</tbody>
</table>
Mount Cloud Volumes ONTAP NFS volume on Linux client

To mount the Cloud Volumes ONTAP (DIY) file system from VMs within Google Cloud VMware Engine, follow the below steps:

Provision the volume following the below steps

1. In the Volumes tab, click Create New Volume.
2. On the Create New Volume page, select a volume type:

   ![Volume Information](image)

   - Disk Type: PD-SSD
   - Tiering Policy: None
   - Capacity: 11.05 GB (10.97 GB Allocated, 0.08 GB Free)

3. In the Volumes tab, place your mouse cursor over the volume, select the menu icon (°), and then click Mount Command.

4. Go to your Linux machine and enter this mount command:

   ```bash
   mount 10.0.6.251:/cvogcvenfsvol01 <dest_dir>
   ```

5. Click Copy.
6. Connect to the designated Linux instance.
7. Open a terminal on the instance using secure shell (SSH) and log in with the appropriate credentials.
8. Make a directory for the volume’s mount point with the following command.
$ sudo mkdir /cvogcvetst

8. Mount the Cloud Volumes ONTAP NFS volume to the directory that is created in the previous step.

```
sudo mount 10.0.6.251:/cvogcvenfsvol01 /cvogcvetst
```

Cloud Volumes Service (CVS)

Cloud Volumes Services (CVS) is a complete portfolio of data services to deliver advanced cloud solutions. Cloud Volumes Services supports multiple file access protocols for major cloud providers (NFS and SMB support).

Other benefits and features include: data protection and restore with Snapshot; special features to replicate, sync and migrate data destinations on-prem or in the cloud; and consistent high performance at the level of a dedicated flash storage system.

Cloud Volumes Service (CVS) as guest connected storage
Configure Cloud Volumes Service with VMware Engine

Cloud Volumes Service shares can be mounted from VMs that are created in the VMware Engine environment. The volumes can also be mounted on the Linux client and mapped on the Windows client because Cloud Volumes Service supports SMB and NFS protocols. Cloud Volumes Service volumes can be set up in simple steps.

Cloud Volume Service and Google Cloud VMware Engine private cloud must be in the same region.

To purchase, enable and configure NetApp Cloud Volumes Service for Google Cloud from the Google Cloud Marketplace, follow this detailed guide.
Create a CVS NFS volume to GCVE private cloud

To create and mount NFS volumes, complete the following steps:

1. Access Cloud Volumes from Partner Solutions within the Google cloud console.

![Google Cloud Console](image)

2. In the Cloud Volumes Console, go to the Volumes page and click Create.

![Cloud Volumes Console](image)

3. On the Create File System page, specify the volume name and billing labels as required for chargeback mechanisms.
4. Select the appropriate service. For GCVE, choose CVS-Performance and desired service level for improved latency and higher performance based on the application workload requirements.

5. Specify the Google Cloud region for the volume and volume path (The volume path must be unique across all of cloud volumes in the project)
6. Select the level of performance for the volume.

7. Specify the size of the volume and the protocol type. In this testing, NFSv3 is used.

8. In this step, select the VPC Network from which the volume will be accessible. Ensure VPC peering is in place.
HINT: If VPC peering has not been done, a pop-up button will be displayed to guide you through the peering commands. Open a Cloud Shell session and execute the appropriate commands to peer your VPC with Cloud Volumes Service producer. In case you decide to prepare VPC peering in beforehand, refer to these instructions.

9. Manage the Export policy rules by adding the appropriate rules and Select the checkbox for the corresponding NFS version.

Note: Access to NFS volumes won’t be possible unless an export policy is added.

10. Click Save to create the volume.
Mounting NFS exports to VMs running on VMware Engine

Before preparing to mount the NFS volume, ensure the peering status of private connection is listed as Active. Once status is Active, use the mount command.

To mount an NFS volume, do the following:

1. In the Cloud Console, go to Cloud Volumes > Volumes.
2. Go to the Volumes page
3. Click the NFS volume for which you want to mount NFS exports.
4. Scroll to the right, under Show More, click Mount Instructions.

To perform the mounting process from within the guest OS of the VMware VM, follow the below steps:

1. Use SSH client and SSH to the virtual machine.
2. Install the nfs client on the instance.
   a. On Red Hat Enterprise Linux or SuSE Linux instance:
      
      ```
      sudo yum install -y nfs-utils
      ```
   b. On an Ubuntu or Debian instance:
      
      ```
      sudo apt-get install nfs-common
      ```
3. Create a new directory on the instance, such as "/nimCVSNFSol01":

      ```
      sudo mkdir /nimCVSNFSol01
      ```
4. Mount the volume using the appropriate command. Example command from the lab is below:

      ```
      sudo mount -t nfs -o rw,hard,rsize=65536,wsize=65536,vers=3,tcp 10.53.0.4:/nimCVSNFSol01 /nimCVSNFSol01
      ```
### Filesystem Usage

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>1K-blocks</th>
<th>Used</th>
<th>Available</th>
<th>Use%</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>udev</td>
<td>16409952</td>
<td>0</td>
<td>16409952</td>
<td>0%</td>
<td>/dev</td>
</tr>
<tr>
<td>tmpfs</td>
<td>16441628</td>
<td>0</td>
<td>16441628</td>
<td>0%</td>
<td>/run</td>
</tr>
<tr>
<td>tmpfs</td>
<td>16441628</td>
<td>0</td>
<td>16441628</td>
<td>0%</td>
<td>/sys/fs/cgroup</td>
</tr>
<tr>
<td>/dev/sdb5</td>
<td>60145932</td>
<td>19231356</td>
<td>38778832</td>
<td>34%</td>
<td>/dev/shm</td>
</tr>
<tr>
<td>/dev/loop0</td>
<td>128</td>
<td>128</td>
<td>0</td>
<td>100%</td>
<td>/snap/barefs</td>
</tr>
<tr>
<td>/dev/loop1</td>
<td>56832</td>
<td>56832</td>
<td>0</td>
<td>100%</td>
<td>/snap/core18/2118</td>
</tr>
<tr>
<td>/dev/loop2</td>
<td>66688</td>
<td>66688</td>
<td>0</td>
<td>100%</td>
<td>/snap/gtk-common-themes/1515</td>
</tr>
<tr>
<td>/dev/loop4</td>
<td>66816</td>
<td>66816</td>
<td>0</td>
<td>100%</td>
<td>/snap/gtk-common-themes/1519</td>
</tr>
<tr>
<td>/dev/loop5</td>
<td>52224</td>
<td>52224</td>
<td>0</td>
<td>100%</td>
<td>/snap/snap-store/547</td>
</tr>
<tr>
<td>/dev/loop8</td>
<td>222756</td>
<td>222756</td>
<td>0</td>
<td>100%</td>
<td>/snap/group-1-3.1-1984/72</td>
</tr>
<tr>
<td>/dev/sdb1</td>
<td>523248</td>
<td>4</td>
<td>523244</td>
<td>1%</td>
<td>/boot/ecl</td>
</tr>
<tr>
<td>tmpfs</td>
<td>3208324</td>
<td>28</td>
<td>3208296</td>
<td>1%</td>
<td>/run/user/1000</td>
</tr>
<tr>
<td>10.53.0.4:/nixCVSNFSol01</td>
<td>107374182400</td>
<td>107374182400</td>
<td>1%</td>
<td>/nixCVSNFSol01</td>
<td></td>
</tr>
</tbody>
</table>
Creating and Mounting SMB Share to VMs running on VMware Engine

For SMB volumes, make sure the Active Directory connections is configured prior to creating the SMB volume.

Once the AD connection is in place, create the volume with the desired service level. The steps are like creating NFS volume except selecting the appropriate protocol.

1. In the Cloud Volumes Console, go to the Volumes page and click Create.
2. On the Create File System page, specify the volume name and billing labels as required for chargeback mechanisms.

3. Select the appropriate service. For GCVE, choose CVS-Performance and desired service level for improved latency and higher performance based on the workload requirements.
4. Specify the Google Cloud region for the volume and volume path (The volume path must be unique across all of cloud volumes in the project)

5. Select the level of performance for the volume.
Create File System

Service Level
Select the performance level required for your workload.

- **Standard**
  - Up to 16 MiB/s per TiB

- **Premium**
  - Up to 64 MiB/s per TiB

- **Extreme**
  - Up to 128 MiB/s per TiB

**Snapshot**
The snapshot to create the volume from.

6. Specify the size of the volume and the protocol type. In this testing, SMB is used.

**Volume Details**

**Allocated Capacity**

- **1024 GiB**

Allocated size must be between 1 TiB (1024 GiB) and 100 TiB (102400 GiB)

**Protocol Type**

- **SMB**

7. In this step, select the VPC Network from which the volume will be accessible. Ensure VPC peering is in place.

**Hints:** If VPC peering has not been done, a pop-up button will be displayed to guide you through the peering commands. Open a Cloud Shell session and execute the appropriate commands to peer your VPC with Cloud Volumes Service producer. In case you decide to prepare VPC peering in
8. Click Save to create the volume.

To mount the SMB volume, do the following:

1. In the Cloud Console, go to Cloud Volumes > Volumes.
2. Go to the Volumes page.
3. Click the SMB volume for which you want to map an SMB share.
4. Scroll to the right, under Show More, click Mount Instructions.

To perform the mounting process from within the Windows guest OS of the VMware VM, follow the below steps:

1. Click the Start button and then click on Computer.
2. Click Map Network Drive.
3. In the Drive list, click any available drive letter.
4. In the folder box, type:
   \nimsmb-3830.nimgcveval.com\nimCVSMBvol01
To connect every time you log on to your computer, select the Reconnect at sign-in check box.

5. Click Finish.

Region Availability for Supplemental NFS datastores on AWS, Azure, and GCP

Learn more about the the Global Region support for supplemental NFS datastores on AWS, Azure and Google Cloud Platform (GCP).

AWS Region Availability

The availability of supplemental NFS datastores on AWS / VMC is defined by Amazon. First, you need to determine if both VMC and FSxN are available in a specified region. Next, you need to determine if the FSxN supplemental NFS datastore is supported in that region.

- Check the availability of VMC [here](#).
- Amazon’s pricing guide offers information on where FSxN (FSx ONTAP) is available. You can find that information [here](#).
- Availability of the FSxN supplemental NFS datastore for VMC is coming soon.

While information is still being released, the following chart identifies the current support for VMC, FSxN and FSxN as a supplemental NFS datastore.
### Americas

<table>
<thead>
<tr>
<th>AWS Region</th>
<th>VMC Availability</th>
<th>FSx ONTAP Availability</th>
<th>NFS Datastore Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (Northern Virginia)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>US East (Ohio)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>US West (Northern California)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>US West (Oregon)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GovCloud (US West)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Canada (Central)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>South America (Sao Paulo)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Last updated on: June 2, 2022.

### EMEA

<table>
<thead>
<tr>
<th>AWS Region</th>
<th>VMC Availability</th>
<th>FSx ONTAP Availability</th>
<th>NFS Datastore Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (Ireland)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (London)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Last updated on: June 2, 2022.

### Asia Pacific

<table>
<thead>
<tr>
<th>AWS Region</th>
<th>VMC Availability</th>
<th>FSx ONTAP Availability</th>
<th>NFS Datastore Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Sydney)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Azure Region Availability

The availability of supplemental NFS datastores on Azure / AVS is defined by Microsoft. First, you need to determine if both AVS and ANF are available in a specific region. Next, you need to determine if the ANF supplemental NFS datastore is supported in that region.

- Check the availability of AVS and ANF here.
- Check the availability of the ANF supplemental NFS datastore here.

GCP Region Availability

GCP region availability will be released when GCP enters public availability.

Summary and Conclusion: Why NetApp Hybrid Multicloud with VMware

NetApp Cloud Volumes along with VMware solutions for the major hyperscalers provides great potential for organizations looking to leverage hybrid cloud. The rest of this section provides the use cases that show integrating NetApp Cloud Volumes enables true hybrid Multicloud capabilities.

Use case #1: Optimizing storage

When performing a sizing exercise using RVtools output, it is always evident that the horsepower (vCPU/vMem) scale is parallel with storage. Many times, organizations find themselves in a situation where the storage space requires drives the size of the cluster well beyond what is needed for horsepower.

By integrating NetApp Cloud Volumes, organizations can realize a vSphere-based cloud solution with a simple migration approach, with no re-platforming, no IP changes, and no architectural changes. Additionally, this optimization enables you to scale the storage footprint while keeping the host count to least amount required in vSphere, but no change to the storage hierarchy, security, or files made available. This allows you to optimize the deployment and reduce the overall TCO by 35–45%. This integration also enables you to scale storage from warm storage to production-level performance in seconds.

Use case #2: Cloud migration

Organizations are under pressure to migrate applications from on-premises data centers to the Public Cloud for multiple reasons: an upcoming lease expiration; a finance directive to move from capital expenditure (capex) spending to operational expenditures (opex) spending; or simply a top-down mandate to move everything to the cloud.

When speed is critical, only a streamlined migration approach is feasible because re-platforming and refactoring applications to adapt to the cloud’s particular IaaS platform is slow and expensive, often taking months. By combining NetApp Cloud Volumes with the bandwidth-efficient SnapMirror replication for guest-connected storage (including RDMs in conjunction with application-consistent Snapshot copies and HCX, cloud specific migration (e.g. Azure Migrate), or third-party products for replicating VMs), this transition is even easier than relying on time-consuming I/O filters mechanisms.

Use case #3: Data center expansion

When a data center reaches capacity limits due to seasonal demand spikes or just steady organic growth,
moving to the cloud-hosted VMware along with NetApp Cloud Volumes is an easy solution. Leveraging NetApp Cloud Volumes allows storage creation, replication, and expansion very easily by providing high availability across availability zones and dynamic scaling capabilities. Leveraging NetApp Cloud Volumes helps in minimizing host cluster capacity by overcoming the need for stretch clusters.

**Use case #4: Disaster recovery to the cloud**

In a traditional approach, if a disaster occurs, the VMs replicated to the cloud would require conversion to the cloud’s own hypervisor platform before they could be restored – not a task to be handled during a crisis.

By using NetApp Cloud Volumes for guest-connected storage using SnapCenter and SnapMirror replication from on-premises along with public cloud virtualization solutions, a better approach for disaster recovery can be devised allowing VM replicas to be recovered on fully consistent VMware SDDC infrastructure along with cloud specific recovery tools (e.g. Azure Site Recovery) or equivalent third-party tools such as Veeam. This approach also enables you to perform disaster recovery drills and recovery from ransomware quickly. This also enables you to scale to full production for testing or during a disaster by adding hosts on-demand.

**Use case #5: Application modernization**

After applications are in the public cloud, organizations will want to take advantage of the hundreds of powerful cloud services to modernize and extend them. With the use of NetApp Cloud Volumes, modernization is an easy process because the application data is not locked into vSAN and allows data mobility for a wide range of use cases, including Kubernetes.

**Conclusion**

Whether you are targeting an all-cloud or hybrid cloud, NetApp Cloud Volumes provides excellent options to deploy and manage the application workloads along with file services and block protocols while reducing the TCO by making the data requirements seamless to the application layer.

Whatever the use case, choose your favorite cloud/hyperscaler together with NetApp Cloud Volumes for rapid realization of cloud benefits, consistent infrastructure, and operations across on-premises and multiple clouds, bidirectional portability of workloads, and enterprise-grade capacity and performance.

It is the same familiar process and procedures that are used to connect the storage. Remember, it is just the position of the data that changed with new names; the tools and processes all remain the same and NetApp Cloud Volumes helps in optimizing the overall deployment.

**VMware Hybrid Cloud Use Cases**

**Use Cases for NetApp Hybrid Multicloud with VMware**

An overview of the use cases of importance to IT organization when planning hybrid-cloud or cloud-first deployments.

**Popular Use Cases**

Use cases include:

- Disaster recovery,
- Hosting workloads during data center maintenance, * quick burst in which additional resources are required beyond what’s provisioned in the local data center,
• VMware site expansion,
• Fast migration to the cloud,
• Dev/test, and
• Modernization of apps leveraging cloud supplemental technologies.

Throughout this documentation, cloud workload references will be detailed using the VMware use-cases. These use-cases are:

• Protect (includes both Disaster Recovery and Backup / Restore)
• Migrate
• Extend

**Inside the IT Journey**

Most organizations are on a journey to transformation and modernization. As part of this process, companies are trying to use their existing VMware investments while leveraging cloud benefits and exploring ways to make the migration process as seamless as possible. This approach would make their modernization efforts very easy because the data is already in the cloud.

The easiest answer to this scenario is VMware offerings in each hyperscaler. Like NetApp® Cloud Volumes, VMware provides a way to move or extend on-premises VMware environments to any cloud, allowing you to retain existing on-premises assets, skills, and tools while running workloads natively in the cloud. This reduces risk because there will be no service breaks or a need for IP changes and provides the IT team the ability to operate the way they do on-premises using existing skills and tools. This can lead to accelerated cloud migrations and a much smoother transition to a hybrid Multicloud architecture.

**Understanding the Importance of Supplemental NFS Storage Options**

While VMware in any cloud delivers unique hybrid capabilities to every customer, limited supplemental NFS storage options have restricted its usefulness for organizations with storage-heavy workloads. Because storage is directly tied to hosts, the only way to scale storage is to add more hosts—and that can increase costs by 35–40 percent or more for storage intensive workloads. These workloads just need additional storage, not additional horsepower. But that means paying for additional hosts.

Let’s consider this scenario:

A customer requires just five hosts for CPU and memory, but has a lot of storage needs, and needs 12 hosts to meet the storage requirement. This requirement ends up really tipping the financial scale by having to buy the additional horsepower, when they only need to increment the storage.

When you’re planning cloud adoption and migrations, it’s always important to evaluate the best approach and take the easiest path that reduces total investments. The most common and easiest approach for any application migration is rehosting (also known as lift and shift) where there is no virtual machine (VM) or data conversion. Using NetApp Cloud Volumes with VMware software-defined data center (SDDC), while complementing vSAN, provides an easy lift-and-shift option.

**NetApp Solutions for Amazon VMware Managed Cloud (VMC)**

Learn more about the solutions that NetApp brings to AWS.

VMware defines the cloud workloads into one of three categories:
• Protect (including both Disaster Recovery and Backup / Restore)
• Migrate
• Extend

Browse the available solutions in the following sections.

Protect
• Disaster Recovery with VMC on AWS (guest connected)
• Veeam Backup & Restore in VMC with FSx for ONTAP
• Disaster Recovery (DRO) with FSx for ONTAP and VMC
• Using Veeam Replication and FSx for ONTAP for Disaster recovery to VMware Cloud on AWS

Migrate
• Migrate Workloads to FSxN datastore using VMware HCX

Extend
COMING SOON!!

NetApp Solutions for Azure VMware Solution (AVS)

Learn more about the solutions that NetApp brings to Azure.

VMware defines the cloud workloads into one of three categories:
• Protect (including both Disaster Recovery and Backup / Restore)
• Migrate
• Extend

Browse the available solutions in the following sections.

Protect
• Disaster Recovery with ANF and JetStream (supplemental NFS datastore)
• Disaster Recovery with ANF and CVO (guest connected storage)
• Disaster Recovery (DRO) with ANF and AVS
• Using Veeam Replication and Azure NetApp Files datastore for disaster recovery to Azure VMware Solution

Migrate
• Migrate Workloads to Azure NetApp Files datastore using VMware HCX

Extend
COMING SOON!!
NetApp Solutions for Google Cloud VMware Engine (GCVE)

Learn more about the solutions that NetApp brings to GCP.

VMware defines the cloud workloads into one of three categories:

- Protect (including both Disaster Recovery and Backup / Restore)
- Migrate
- Extend

Browse the available solutions in the following sections.

**Protect**
- Application Disaster Recovery with SnapCenter, Cloud Volumes ONTAP and Veeam Replication
- Application Consistent Disaster Recovery with NetApp SnapCenter and Veeam Replication to NetApp CVS on GCVE

**Migrate**
- Workload Migration using VMware HCX to NetApp Cloud Volume Service NFS datastore
- VM Replication using Veeam to NetApp Cloud Volume Service NFS datastore

**Extend**
COMING SOON!!

NetApp Hybrid Multicloud with VMware Solutions

Protecting Workloads on AWS / VMC

TR-4931: Disaster Recovery with VMware Cloud on Amazon Web Services and Guest Connect

Authors: Chris Reno, Josh Powell, and Suresh Thoppay - NetApp Solutions Engineering

Overview

A proven disaster recovery (DR) environment and plan is critical for organizations to ensure that business-critical applications can be rapidly restored in the event of a major outage. This solution focuses on demonstrating DR use cases with a focus on VMware and NetApp technologies, both on-premises and with VMware Cloud on AWS.

NetApp has a long history of integration with VMware as evidenced by the tens of thousands of customers that have chosen NetApp as their storage partner for their virtualized environment. This integration continues with guest-connected options in the cloud and recent integrations with NFS datastores as well. This solution focuses on the use case commonly referred to as guest-connected storage.

In guest-connected storage, the guest VMDK is deployed on a VMware-provisioned datastore, and application data is housed on iSCSI or NFS and mapped directly to the VM. Oracle and MS SQL applications are used to demonstrate a DR scenario, as shown in the following figure.
Assumptions, pre-requisites and component overview

Before deploying this solution, review the overview of the components, the required pre-requisites to deploy the solution and assumptions made in documenting this solution.

DR Solution Requirements, Pre-requisites and Planning

Performing DR with SnapCenter

In this solution, SnapCenter provides application-consistent snapshots for SQL Server and Oracle application data. This configuration, together with SnapMirror technology, provides high-speed data replication between our on-premises AFF and FSx ONTAP cluster. Additionally, Veeam Backup & Replication provides backup and restore capabilities for our virtual machines.

In this section, we cover the configuration of SnapCenter, SnapMirror, and Veeam for both backup and restore.

The following sections cover configuration and the steps needed to complete a failover at the secondary site:

Configure SnapMirror relationships and retention schedules

SnapCenter can update SnapMirror relationships within the primary storage system (primary > mirror) and to secondary storage systems (primary > vault) for the purpose of long-term archiving and retention. To do so, you must establish and initialize a data replication relationship between a destination volume and a source volume using SnapMirror.

The source and destination ONTAP systems must be in networks that are peered using Amazon VPC peering, a transit gateway, AWS Direct Connect, or an AWS VPN.

The following steps are required for setting up SnapMirror relationships between an on-premises ONTAP system and FSx ONTAP:

Refer to the FSx for ONTAP – ONTAP User Guide for more information on creating SnapMirror relationships with FSx.
Record the source and destination Intercluster logical interfaces

For the source ONTAP system residing on-premises, you can retrieve the inter-cluster LIF information from System Manager or from the CLI.

1. In ONTAP System Manager, navigate to the Network Overview page and retrieve the IP addresses of Type: Intercluster that are configured to communicate with the AWS VPC where FSx is installed.

2. To retrieve the Intercluster IP addresses for FSx, log into the CLI and run the following command:

   FSx-Dest::> network interface show -role intercluster

   FsxId0ae40e08acc0dea67::> network interface show -role intercluster

   +-----------------+-------------------+---------------------+-------------------+-------------------+-------------------+------------------+
   | Logical Vserver | Interface         | Admin/Oper Status   | Network Address/Netmask | Current Node     | Current Is Port   | Current Is Home  |
   +-----------------+-------------------+---------------------+---------------------+-------------------+-------------------+------------------+
   | FsxId0ae40e08acc0dea67 | inter_1          | up/up               | 172.30.15.42/25      | FsxId0ae40e08acc0dea67-01 | e0e              | true             |
   | FsxId0ae40e08acc0dea67 | inter_2          | up/up               | 172.30.14.28/26      | FsxId0ae40e08acc0dea67-02 | e0e              | true             |
   +-----------------+-------------------+---------------------+---------------------+-------------------+-------------------+------------------+
   2 entries were displayed.
Establish cluster peering between ONTAP and FSx

To establish cluster peering between ONTAP clusters, a unique passphrase entered at the initiating ONTAP cluster must be confirmed in the other peer cluster.

1. Set up peering on the destination FSx cluster using the `cluster peer create` command. When prompted, enter a unique passphrase that is used later on the source cluster to finalize the creation process.

   ```sh
   FSx-Dest::> cluster peer create -address-family ipv4 -peer-addrs source_intercluster_1, source_intercluster_2
   Enter the passphrase:
   Confirm the passphrase:
   ```

2. At the source cluster, you can establish the cluster peer relationship using either ONTAP System Manager or the CLI. From ONTAP System Manager, navigate to Protection > Overview and select Peer Cluster.
3. In the Peer Cluster dialog box, fill out the required information:
   a. Enter the passphrase that was used to establish the peer cluster relationship on the destination FSx cluster.
b. Select **Yes** to establish an encrypted relationship.

c. Enter the intercluster LIF IP address(es) of the destination FSx cluster.

d. Click Initiate Cluster Peering to finalize the process.

4. Verify the status of the cluster peer relationship from the FSx cluster with the following command:

```bash
FSx-Dest::> cluster peer show
```

```
FsxId0ae40e08acc0dea67::> cluster peer show
Peer Cluster Name       Cluster Serial Number       Availability     Authentication
--------------------------------------------------------------------------------------------
E13A300                 1-80-000011                Available           ok
```
Establish SVM peering relationship

The next step is to set up an SVM relationship between the destination and source storage virtual machines that contain the volumes that will be in SnapMirror relationships.

1. From the source FSx cluster, use the following command from the CLI to create the SVM peer relationship:

   ```bash
   FSx-Dest::> vserver peer create -vserver DestSVM -peer-vserver Backup -peer-cluster OnPremSourceSVM -applications snapmirror
   ```

2. From the source ONTAP cluster, accept the peering relationship with either ONTAP System Manager or the CLI.

3. From ONTAP System Manager, go to Protection > Overview and select Peer Storage VMs under Storage VM Peers.

4. In the Peer Storage VM’s dialog box, fill out the required fields:
   - The source storage VM
   - The destination cluster
   - The destination storage VM
5. Click Peer Storage VMs to complete the SVM peering process.
Create a snapshot retention policy

SnapCenter manages retention schedules for backups that exist as snapshot copies on the primary storage system. This is established when creating a policy in SnapCenter. SnapCenter does not manage retention policies for backups that are retained on secondary storage systems. These policies are managed separately through a SnapMirror policy created on the secondary FSx cluster and associated with the destination volumes that are in a SnapMirror relationship with the source volume.

When creating a SnapCenter policy, you have the option to specify a secondary policy label that is added to the SnapMirror label of each snapshot generated when a SnapCenter backup is taken.

On the secondary storage, these labels are matched to policy rules associated with the destination volume for the purpose of enforcing retention of snapshots.

The following example shows a SnapMirror label that is present on all snapshots generated as part of a policy used for daily backups of our SQL Server database and log volumes.

For more information on creating SnapCenter policies for a SQL Server database, see the SnapCenter documentation.

You must first create a SnapMirror policy with rules that dictate the number of snapshot copies to retain.

1. Create the SnapMirror Policy on the FSx cluster.

   ```bash
   FSx-Dest::> snapmirror policy create -vserver DestSVM -policy PolicyName -type mirror-vault -restart always
   ```

2. Add rules to the policy with SnapMirror labels that match the secondary policy labels specified in the SnapCenter policies.

   ```bash
   FSx-Dest::> snapmirror policy add-rule -vserver DestSVM -policy PolicyName -snapmirror-label SnapMirrorLabelName -keep #ofSnapshotsToRetain
   ```

The following script provides an example of a rule that could be added to a policy:
Create additional rules for each SnapMirror label and the number of snapshots to be retained (retention period).

Create destination volumes

To create a destination volume on FSx that will be the recipient of snapshot copies from our source volumes, run the following command on FSx ONTAP:

```
FSx-Dest::> volume create -vserver DestSVM -volume DestVolName -aggregate DestAggrName -size VolSize -type DP
```

Create the SnapMirror relationships between source and destination volumes

To create a SnapMirror relationship between a source and destination volume, run the following command on FSx ONTAP:

```
FSx-Dest::> snapmirror create -source-path OnPremSourceSVM:OnPremSourceVol -destination-path DestSVM:DestVol -type XDP -policy PolicyName
```

Initialize the SnapMirror relationships

Initialize the SnapMirror relationship. This process initiates a new snapshot generated from the source volume and copies it to the destination volume.

```
FSx-Dest::> snapmirror initialize -destination-path DestSVM:DestVol
```

Deploy and configure Windows SnapCenter server on-premises.
Deploy Windows SnapCenter Server on premises

This solution uses NetApp SnapCenter to take application-consistent backups of SQL Server and Oracle databases. In conjunction with Veeam Backup & Replication for backing up virtual machine VMDKs, this provides a comprehensive disaster recovery solution for on-premises and cloud-based datacenters.

SnapCenter software is available from the NetApp support site and can be installed on Microsoft Windows systems that reside either in a domain or workgroup. A detailed planning guide and installation instructions can be found at the NetApp Documentation Center.

The SnapCenter software can be obtained at this link.

After it is installed, you can access the SnapCenter console from a web browser using https://Virtual_Cluster_IP_or_FQDN:8146.

After you log into the console, you must configure SnapCenter for backup SQL Server and Oracle databases.
Add storage controllers to SnapCenter

To add storage controllers to SnapCenter, complete the following steps:

1. From the left menu, select Storage Systems and then click New to begin the process of adding your storage controllers to SnapCenter.

![SnapCenter interface showing ONTAP Storage Connections](image)

2. In the Add Storage System dialog box, add the management IP address for the local on-premises ONTAP cluster and the username and password. Then click Submit to begin discovery of the storage system.
3. Repeat this process to add the FSx ONTAP system to SnapCenter. In this case, select More Options at the bottom of the Add Storage System window and click the check box for Secondary to designate the FSx system as the secondary storage system updated with SnapMirror copies or our primary backup snapshots.
For more information related to adding storage systems to SnapCenter, see the documentation at this link.
Add hosts to SnapCenter

The next step is adding host application servers to SnapCenter. The process is similar for both SQL Server and Oracle.

1. From the left menu, select Hosts and then click Add to begin the process of adding storage controllers to SnapCenter.

2. In the Add Hosts window, add the Host Type, Hostname, and the host system Credentials. Select the plug-in type. For SQL Server, select the Microsoft Windows and Microsoft SQL Server plug-in.

3. For Oracle, fill out the required fields in the Add Host dialog box and select the check box for the Oracle Database plug-in. Then click Submit to begin the discovery process and to add the host to SnapCenter.
Add Host

Host Type: Linux
Host Name: oraclesrv_11.sddc.netapp.com
Credentials: root

Select Plug-ins to Install: SnapCenter Plug-ins Package 4.6 for Linux

- [x] Oracle Database
- [ ] SAP HANA

More Options: Port, Install Path, Custom Plug-Ins...

Submit | Cancel
Create SnapCenter policies

Policies establish the specific rules to be followed for a backup job. They include, but are not limited to, the backup schedule, replication type, and how SnapCenter handles backing up and truncating transaction logs.

You can access policies in the Settings section of the SnapCenter web client.

For complete information on creating policies for SQL Server backups, see the SnapCenter documentation.

For complete information on creating policies for Oracle backups, see the SnapCenter documentation.

Notes:

- As you progress through the policy creation wizard, take special note of the Replication section. In this section you stipulate the types of secondary SnapMirror copies that you want taken during the backups process.
- The “Update SnapMirror after creating a local Snapshot copy” setting refers to updating a SnapMirror relationship when that relationship exists between two storage virtual machines residing on the same cluster.
- The “Update SnapVault after creating a local Snapshot copy” setting is used to update a SnapMirror relationship that exists between two separate cluster and between an on-premises ONTAP system and Cloud Volumes ONTAP or FSxN.

The following image shows the preceding options and how they look in the backup policy wizard.
Create SnapCenter Resource Groups

Resource Groups allow you to select the database resources you want to include in your backups and the policies followed for those resources.

1. Go to the Resources section in the left-hand menu.
2. At the top of the window, select the resource type to work with (In this case Microsoft SQL Server) and then click New Resource Group.

The SnapCenter documentation covers step-by-step details for creating Resource Groups for both SQL Server and Oracle databases.

For backing up SQL resources, follow [this link](#).

For Backing up Oracle resources, follow [this link](#).
Deploy and configure Veeam Backup Server

Veeam Backup & Replication software is used in the solution to back up our application virtual machines and archive a copy of the backups to an Amazon S3 bucket using a Veeam scale-out backup repository (SOBR). Veeam is deployed on a Windows server in this solution. For specific guidance on deploying Veeam, see the Veeam help Center Technical documentation.
Configure Veeam scale-out backup repository

After you deploy and license the software, you can create a scale-out backup repository (SOBR) as target storage for backup jobs. You should also include an S3 bucket as a backup of VM data offsite for disaster recovery.

See the following prerequisites before getting started.

1. Create an SMB file share on your on-premises ONTAP system as the target storage for backups.
2. Create an Amazon S3 bucket to include in the SOBR. This is a repository for the offsite backups.
Add ONTAP Storage to Veeam

First, add the ONTAP storage cluster and associated SMB/NFS filesystem as storage infrastructure in Veeam.

1. Open the Veeam console and log in. Navigate to Storage Infrastructure and then select Add Storage.

2. In the Add Storage wizard, select NetApp as the storage vendor and then select Data ONTAP.

3. Enter the management IP address and check the NAS Filer box. Click Next.
4. Add your credentials to access the ONTAP cluster.

5. On the NAS Filer page choose the desired protocols to scan and select Next.
6. Complete the Apply and Summary pages of the wizard and click Finish to begin the storage discovery process. After the scan completes, the ONTAP cluster is added along with the NAS filers as available resources.

7. Create a backup repository using the newly discovered NAS shares. From Backup Infrastructure, select Backup Repositories and click the Add Repository menu item.
8. Follow all steps in the New Backup Repository Wizard to create the repository. For detailed information on creating Veeam Backup Repositories, see the Veeam documentation.
New Backup Repository

**Share**
Type in UNC path to share (mapped drives are not supported), specify share access credentials and how backup jobs should write data to this share.

**Name**

**Share**

**Repository**

**Mount Server**

**Review**

**Apply**

**Summary**

**Shared Folder:**
```
\\172.21.162.18\\VBRRepo
```
**Use \server\folder format**

- **This share requires access credentials:**
  - \sddc\administrator (sddc\administrator, last edited: 85 days ago)
  - **Add...**

**Gateway server:**
- **Automatic selection**
- **The following server:**
  - veerm.sddc.netapp.com (Backup server)

Use this option to improve performance and reliability of backup to a NAS located in a remote site.
Add the Amazon S3 bucket as a backup repository

The next step is to add the Amazon S3 storage as a backup repository.

1. Navigate to Backup Infrastructure > Backup Repositories. Click Add Repository.

2. In the Add Backup Repository wizard, select Object Storage and then Amazon S3. This starts the New Object Storage Repository wizard.
Add Backup Repository
Select the type of backup repository you want to add.

1. Direct attached storage
Microsoft Windows or Linux server with internal or direct attached storage. This configuration enables data movers to run directly on the server, allowing for fastest performance.

2. Network attached storage
Network share on a file server or a NAS device. When backing up to a remote share, we recommend that you select a gateway server located in the same site with the share.

3. Deduplicating storage appliance
Dell EMC Data Domain, EitaGrid, HPE StoreOnce or Quantum DXL. If you are unable to meet the requirements of advanced integration via native appliance API, use the network attached storage option instead.

4. Object storage
On-prem object storage system or a cloud object storage provider. Object storage can only be used as a Capacity Tier of scale-out backup repositories. Backing up directly to object storage is not currently supported.

3. Provide a name for your object storage repository and click Next.
4. In the next section, provide your credentials. You need an AWS Access Key and Secret Key.

- **Account**
  - Specify AWS account to use for connecting to Amazon S3 storage bucket.

5. After the Amazon configuration loads, choose your datacenter, bucket, and folder and click Apply. Finally, click Finish to close out the wizard.
Now that we have added our storage repositories to Veeam, we can create the SOBR to automatically tier backup copies to our offsite Amazon S3 object storage for disaster recovery.

1. From Backup Infrastructure, select Scale-out Repositories and then click the Add Scale-out Repository menu item.

2. In the New Scale-out Backup Repository provide a name for the SOBR and click Next.

3. For the Performance Tier, choose the backup repository that contains the SMB share residing on your local ONTAP cluster.
4. For the Placement Policy, choose either Data Locality or Performance based your requirements. Select next.

5. For Capacity Tier we extend the SOBR with Amazon S3 object storage. For the purposes of disaster recovery, select Copy Backups to Object Storage as Soon as They are Created to ensure timely delivery of our secondary backups.

6. Finally, select Apply and Finish to finalize creation of the SOBR.

Create the scale-out backup repository jobs

The final step to configuring Veeam is to create backup jobs using the newly created SOBR as the backup destination. Creating backup jobs is a normal part of any storage administrator’s repertoire and we do not cover the detailed steps here. For more complete information on creating backup jobs in Veeam, see the Veeam Help Center Technical Documentation.
BlueXP backup and recovery tools and configuration

To conduct a failover of application VMs and database volumes to VMware Cloud Volume services running in AWS, you must install and configure a running instance of both SnapCenter Server and Veeam Backup and Replication Server. After the failover is complete, you must also configure these tools to resume normal backup operations until a failback to the on-premises datacenter is planned and executed.

Deploy secondary Windows SnapCenter Server

SnapCenter Server is deployed in the VMware Cloud SDDC or installed on an EC2 instance residing in a VPC with network connectivity to the VMware Cloud environment.

SnapCenter software is available from the NetApp support site and can be installed on Microsoft Windows systems that reside either in a domain or workgroup. A detailed planning guide and installation instructions can be found at the NetApp documentation center.

You can find the SnapCenter software at this link.

Configure secondary Windows SnapCenter Server

To perform a restore of application data mirrored to FSx ONTAP, you must first perform a full restore of the on-premises SnapCenter database. After this process is complete, communication with the VMs is reestablished and application backups can now resume using FSx ONTAP as the primary storage.

To achieve this, you must complete the following items on the SnapCenter Server:

1. Configure the computer name to be identical to the original on-premises SnapCenter Server.
2. Configure networking to communicate with VMware Cloud and the FSx ONTAP instance.
3. Complete the procedure to restore the SnapCenter database.
4. Confirm that SnapCenter is in Disaster Recovery mode to make sure that FSx is now the primary storage for backups.
5. Confirm that communication is reestablished with the restored virtual machines.

For more information on completing these steps, see to section "SnapCenter database Restore Process".

Deploy secondary Veeam Backup & Replication server

You can install the Veeam Backup & Replication server on a Windows server in the VMware Cloud on AWS or on an EC2 instance. For detailed implementation guidance, see the Veeam Help Center Technical Documentation.
Configure secondary Veeam Backup & Replication server

To perform a restore of virtual machines that have been backed up to Amazon S3 storage, you must install the Veeam Server on a Windows server and configure it to communicate with VMware Cloud, FSx ONTAP, and the S3 bucket that contains the original backup repository. It must also have a new backup repository configured on FSx ONTAP to conduct new backups of the VMs after they are restored.

To perform this process, the following items must be completed:

1. Configure networking to communicate with VMware Cloud, FSx ONTAP, and the S3 bucket containing the original backup repository.
2. Configure an SMB share on FSx ONTAP to be a new backup repository.
3. Mount the original S3 bucket that was used as part of the scale-out backup repository on premises.
4. After restoring the VM, establish new backup jobs to protect SQL and Oracle VMs.

For more information on restoring VMs using Veeam, see the section "Restore Application VMs with Veeam Full Restore".

SnapCenter database backup for disaster recovery

SnapCenter allows for the backup and recovery of its underlying MySQL database and configuration data for the purpose of recovering the SnapCenter server in the case of a disaster. For our solution, we recovered the SnapCenter database and configuration on an AWS EC2 instance residing in our VPC. For more information on this step, see this link.

SnapCenter backup prerequisites

The following prerequisites are required for SnapCenter backup:

- A volume and SMB share created on the on-premises ONTAP system to locate the backed-up database and configuration files.
- A SnapMirror relationship between the on-premises ONTAP system and FSx or CVO in the AWS account. This relationship is used for transporting the snapshot containing the backed-up SnapCenter database and configuration files.
- Windows Server installed in the cloud account, either on an EC2 instance or on a VM in the VMware Cloud SDDC.
- SnapCenter installed on the Windows EC2 instance or VM in VMware Cloud.
SnapCenter backup and restore process summary

- Create a volume on the on-premises ONTAP system for hosting the backup db and config files.
- Set up a SnapMirror relationship between on-premises and FSx/CVO.
- Mount the SMB share.
- Retrieve the Swagger authorization token for performing API tasks.
- Start the db restore process.
- Use the xcopy utility to copy the db and config file local directory to the SMB share.
- On FSx, create a clone of the ONTAP volume (copied via SnapMirror from on-premises).
- Mount the SMB share from FSx to EC2/VMware Cloud.
- Copy the restore directory from the SMB share to a local directory.
- Run the SQL Server restore process from Swagger.
Back up the SnapCenter database and configuration

SnapCenter provides a web client interface for executing REST API commands. For information on accessing the REST APIs through Swagger, see the SnapCenter documentation at this link.
Log into Swagger and obtain authorization token

After you have navigated to the Swagger page, you must retrieve an authorization token to initiate the database restore process.


2. Expand the Auth section and click Try it Out.

3. In the UserOperationContext area, fill in the SnapCenter credentials and role and click Execute.
4. In the Response body below, you can see the token. Copy the token text for authentication when executing the backup process.
Perform a SnapCenter database backup

Next go to the Disaster Recovery area on the Swagger page to begin the SnapCenter backup process.

1. Expand the Disaster Recovery area by clicking it.

2. Expand the `/4.6/disasterrecovery/server/backup` section and click Try it Out.

3. In the SmDRBackupRequest section, add the correct local target path and select Execute to start the backup of the SnapCenter database and configuration.

   The backup process does not allow backing up directly to an NFS or CIFS file share.
Token * required
string
User authorization token
TUHFUMD69XRe5cuW9nwyj4b0i5Y5FN3XOjQ==

SmDRBackupRequest * required
object
Parameters to take Backup

Edit Value | Model
---|---

```
{
  "TargetPath": "C:\SnapCenter_Backups\"
}
```

Parameter content type
application/json

Execute
Monitor the backup job from SnapCenter

Log into SnapCenter to review log files when starting the database restore process. Under the Monitor section, you can view the details of the SnapCenter server disaster recovery backup.

---

Use XCOPY utility to copy the database backup file to the SMB share

Next you must move the backup from the local drive on the SnapCenter server to the CIFS share that is used to SnapMirror copy the data to the secondary location located on the FSx instance in AWS. Use xcopy with specific options that retain the permissions of the files.

Open a command prompt as Administrator. From the command prompt, enter the following commands:

```bash
xcopy  <Source_Path>  \\
\<Destination_Server_IP>\<Folder_Path>  /O /X /E /H /K
xcopy c:\SC_Backups\SnapCenter_DR  \10.61.181.185\snapcenter_dr  /O /X /E /H /K
```

Failover

Disaster occurs at primary site

For a disaster that occurs at the primary on-premises datacenter, our scenario includes failover to a secondary site residing on Amazon Web Services infrastructure using VMware Cloud on AWS. We assume that the virtual machines and our on-premises ONTAP cluster are no longer accessible. In addition, both the SnapCenter and Veeam virtual machines are no longer accessible and must be rebuilt at our secondary site.

This section address failover of our infrastructure to the cloud, and we cover the following topics:

- SnapCenter database restore. After a new SnapCenter server has been established, restore the MySQL database and configuration files and toggle the database into disaster recovery mode in order to allow the secondary FSx storage to become the primary storage device.
- Restore the application virtual machines using Veeam Backup & Replication. Connect the S3 storage that contains the VM backups, import the backups, and restore them to VMware Cloud on AWS.
- Restore the SQL Server application data using SnapCenter.
- Restore the Oracle application data using SnapCenter.
SnapCenter database restore process

SnapCenter supports disaster recovery scenarios by allowing the backup and restore of its MySQL database and configuration files. This allows an administrator to maintain regular backups of the SnapCenter database at the on-premises datacenter and later restore that database to a secondary SnapCenter database.

To access the SnapCenter backup files on the remote SnapCenter server, complete the following steps:

1. Break the SnapMirror relationship from the FSx cluster, which makes the volume read/write.
2. Create a CIFS server (if necessary) and create a CIFS share pointing to the junction path of the cloned volume.
3. Use xcopy to copy the backup files to a local directory on the secondary SnapCenter system.
4. Install SnapCenter v4.6.
5. Ensure that SnapCenter server has the same FQDN as the original server. This is required for the db restore to be successful.

To start the restore process, complete the following steps:

1. Navigate to the Swagger API web page for the secondary SnapCenter server and follow the previous instructions to obtain an authorization token.
2. Navigate to the Disaster Recovery section of the Swagger page, select /4.6/disasterrecovery/server/restore, and click Try it Out.

3. Paste in your authorization token and, in the SmDRResterRequest section, paste in the name of the backup and the local directory on the secondary SnapCenter server.

4. Select the Execute button to start the restore process.
5. From SnapCenter, navigate to the Monitor section to view the progress of the restore job.

6. To enable SQL Server restores from secondary storage, you must toggle the SnapCenter database into Disaster Recovery mode. This is performed as a separate operation and initiated on the Swagger API web page.
   
a. Navigate to the Disaster Recovery section and click `/4.6/disasterrecovery/storage`.
   
b. Paste in the user authorization token.
   
c. In the SmSetDisasterRecoverySettingsRequest section, change `EnableDisasterRecover` to `true`.
   
d. Click Execute to enable disaster recovery mode for SQL Server.
See comments regarding additional procedures.

Restore application VMs with Veeam full restore
Create a backup repository and import backups from S3

From the secondary Veeam server, import the backups from S3 storage and restore the SQL Server and Oracle VMs to your VMware Cloud cluster.

To import the backups from the S3 object that was part of the on-premises scale-out backup repository, complete the following steps:

1. Go to Backup Repositories and click Add Repository in the top menu to launch the Add Backup Repository wizard. On the first page of the wizard, select Object Storage as the backup repository type.

2. Select Amazon S3 as the Object Storage type.
3. From the list of Amazon Cloud Storage Services, select Amazon S3.

4. Select your pre-entered credentials from the drop-down list or add a new credential for accessing the cloud storage resource. Click Next to continue.
5. On the Bucket page, enter the data center, bucket, folder, and any desired options. Click Apply.
6. Finally, select Finish to complete the process and add the repository.
Import backups from S3 object storage

To import the backups from the S3 repository that was added in the previous section, complete the following steps.

1. From the S3 backup repository, select Import Backups to launch the Import Backups wizard.

![Backup Infrastructure](image)

2. After the database records for the import have been created, select Next and then Finish at the summary screen to start the import process.

![Import Backups](image)

3. After the import is complete, you can restore VMs into the VMware Cloud cluster.
<table>
<thead>
<tr>
<th>Message</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔ Starting backup repositories synchronization</td>
<td></td>
</tr>
<tr>
<td>✔ Enumerating repositories</td>
<td></td>
</tr>
<tr>
<td>✔ Found 1 repository</td>
<td></td>
</tr>
<tr>
<td>✔ Processing capacity tier extent of S3 Backup Repository 2</td>
<td>0:03:23</td>
</tr>
<tr>
<td>✔ S3 Backup Repository: added 2 unencrypted</td>
<td>0:03:20</td>
</tr>
<tr>
<td>✔ Importing backup 2 out of 2</td>
<td>0:03:15</td>
</tr>
<tr>
<td>✔ Backup repositories synchronization completed successfully</td>
<td></td>
</tr>
</tbody>
</table>
To restore SQL and Oracle virtual machines to the VMware Cloud on AWS workload domain/cluster, complete the following steps.

1. From the Veeam Home page, select the object storage containing the imported backups, select the VMs to restore, and then right click and select Restore Entire VM.

2. On the first page of the Full VM Restore wizard, modify the VMs to backup if desired and select Next.
3. On the Restore Mode page, select Restore to a New Location, or with Different Settings.
4. On the host page, select the Target ESXi host or cluster to restore the VM to.
5. On the Datastores page, select the target datastore location for both the configuration files and hard disk.
6. On the Network page, map the original networks on the VM to the networks in the new target location.
Network

By default, we will connect the restored VM to the same virtual networks as the original VM. If you are restoring to a different location, specify how networks map between original and new locations.

Network connections:

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQLSRV-04</td>
<td>Not connected</td>
</tr>
<tr>
<td>Management 181 (DSwitch)</td>
<td>Not connected</td>
</tr>
<tr>
<td>Data - A - 3374 (DSwitch)</td>
<td>Not connected</td>
</tr>
<tr>
<td>Data - B - 3375 (DSwitch)</td>
<td>Not connected</td>
</tr>
</tbody>
</table>

Select multiple VMs to apply settings change in bulk.
7. Select whether to scan the restored VM for malware, review the summary page, and click Finish to start the restore.

**Restore SQL Server application data**

The following process provides instructions on how to recover a SQL Server in VMware Cloud Services in AWS in the event of a disaster that renders the on-premises site inoperable.

The following prerequisites are assumed to be complete in order to continue with the recovery steps:

1. The Windows Server VM has been restored to the VMware Cloud SDDC using Veeam Full Restore.
2. A secondary SnapCenter server has been established and SnapCenter database restore and configuration has been completed using the steps outlined in the section "SnapCenter backup and restore process summary."
VM: Post restore configuration for SQL Server VM

After the restore of the VM is complete, you must configure networking and other items in preparation for rediscovering the host VM within SnapCenter.

1. Assign new IP addresses for Management and iSCSI or NFS.
2. Join the host to the Windows domain.
3. Add the hostnames to DNS or to the hosts file on the SnapCenter server.

If the SnapCenter plug-in was deployed using domain credentials different than the current domain, you must change the Log On account for the Plug-in for Windows Service on the SQL Server VM. After changing the Log On account, restart the SnapCenter SMCore, Plug-in for Windows, and Plug-in for SQL Server services.

To automatically rediscover the restored VMs in SnapCenter, the FQDN must be identical to the VM that was originally added to the SnapCenter on premises.

Configure FSx storage for SQL Server restore

To accomplish the disaster recovery restore process for a SQL Server VM, you must break the existing SnapMirror relationship from the FSx cluster and grant access to the volume. To do so, complete the following steps.

1. To break the existing SnapMirror relationship for the SQL Server database and log volumes, run the following command from the FSx CLI:

   ```bash
   FSx-Dest::> snapmirror break -destination-path DestSVM:DestVolName
   ```

2. Grant access to the LUN by creating an initiator group containing the iSCSI IQN of the SQL Server Windows VM:

   ```bash
   FSx-Dest::> igroup create -vserver DestSVM -igroup igroupName -protocol iSCSI -ostype windows -initiator IQN
   ```

3. Finally, map the LUNs to the initiator group that you just created:

   ```bash
   FSx-Dest::> lun mapping create -vserver DestSVM -path LUNPath igroup igroupName
   ```

4. To find the path name, run the `lun show` command.
Set up the Windows VM for iSCSI access and discover the file systems

1. From the SQL Server VM, set up your iSCSI network adapter to communicate on the VMware Port Group that has been established with connectivity to the iSCSI target interfaces on your FSx instance.

2. Open the iSCSI Initiator Properties utility and clear out the old connectivity settings on the Discovery, Favorite Targets, and Targets tabs.

3. Locate the IP address(es) for accessing the iSCSI logical interface on the FSx instance/cluster. This can be found in the AWS console under Amazon FSx > ONTAP > Storage Virtual Machines.

4. From the Discovery tab, click Discover Portal and enter the IP addresses for your FSx iSCSI targets.
5. On the Target tab, click Connect, select Enable Multi-Path if appropriate for your configuration and then click OK to connect to the target.
6. Open the Computer Management utility and bring the disks online. Verify that they retain the same drive letters that they previously held.
Attach the SQL Server databases

1. From the SQL Server VM, open Microsoft SQL Server Management Studio and select Attach to start the process of connecting to the database.

2. Click Add and navigate to the folder containing the SQL Server primary database file, select it, and click OK.
3. If the transaction logs are on a separate drive, choose the folder that contains the transaction log.

4. When finished, click OK to attach the database.
Confirm SnapCenter communication with SQL Server Plug-in

With the SnapCenter database restored to its previous state, it automatically redisCOVERs the SQL Server hosts. For this to work correctly, keep in mind the following prerequisites:

- SnapCenter must be placed in Disaster Recover mode. This can be accomplished through the Swagger API or in Global Settings under Disaster Recovery.
- The FQDN of the SQL Server must be identical to the instance that was running in the on-premises datacenter.
- The original SnapMirror relationship must be broken.
- The LUNs containing the database must be mounted to the SQL Server instance and the database attached.

To confirm that SnapCenter is in Disaster Recovery mode, navigate to Settings from within the SnapCenter web client. Go to the Global Settings tab and then click Disaster Recovery. Make sure that the Enable Disaster Recovery checkbox is enabled.
Restore Oracle application data

The following process provides instructions on how to recover Oracle application data in VMware Cloud Services in AWS in the event of a disaster that renders the on-premises site inoperable.

Complete the following prerequisites to continue with the recovery steps:

1. The Oracle Linux server VM has been restored to the VMware Cloud SDDC using Veeam Full Restore.
2. A secondary SnapCenter server has been established and the SnapCenter database and configuration files have been restored using the steps outlined in this section "SnapCenter backup and restore process summary."
Configure FSx for Oracle restore – Break the SnapMirror relationship

To make the secondary storage volumes hosted on the FSxN instance accessible to the Oracle servers, you must first break the existing SnapMirror relationship.

1. After logging into the FSx CLI, run the following command to view the volumes filtered by the correct name.

   ```
   FSx-Dest::> volume show -volume VolumeName*
   ```

2. Run the following command to break the existing SnapMirror relationships.

   ```
   FSx-Dest::> snapmirror break -destination-path DestSVM:DestVolName
   ```

3. Update the junction-path in the Amazon FSx web client:
4. Add the junction path name and click Update. Specify this junction path when mounting the NFS volume from the Oracle server.
Update volume

Junction path
/oraclesrv_03_u01_dest

The location within your file system where your volume will be mounted.

Volume size
102400
Minimum 20 MiB; Maximum 104857600 MiB

Storage efficiency
Select whether you would like to enable ONTAP storage efficiencies on your volume: deduplication, compression, and compaction.

- Enabled (recommended)
- Disabled

Capacity pool tiering policy
You can optionally enable automatic tiering of your data to lower-cost capacity pool storage.

- Snapshot Only

Cancel  Update
Mount NFS volumes on Oracle Server

In Cloud Manager, you can obtain the mount command with the correct NFS LIF IP address for mounting the NFS volumes that contain the Oracle database files and logs.

1. In Cloud Manager, access the list of volumes for your FSx cluster.

2. From the action menu, select Mount Command to view and copy the mount command to be used on our Oracle Linux server.
3. Mount the NFS file system to the Oracle Linux Server. The directories for mounting the NFS share already exist on the Oracle Linux host.

4. From the Oracle Linux server, use the mount command to mount the NFS volumes.
FSx-Dest::> mount -t oracle_server_ip:/junction-path

Repeat this step for each volume associated with the Oracle databases.

To make the NFS mount persistent upon rebooting, edit the `/etc/fstab` file to include the mount commands.

5. Reboot the Oracle server. The Oracle databases should start up normally and be available for use.

Failback

Upon successful completion of the failover process outlined in this solution, SnapCenter and Veeam resume their backup functions running in AWS, and FSx for ONTAP is now designated as primary storage with no existing SnapMirror relationships with the original on-premises datacenter. After normal function has resumed on premises, you can use a process identical to the one outlined in this documentation to mirror data back to the on-premises ONTAP storage system.

As is also outlined in this documentation, you can configure SnapCenter to mirror the application data volumes from FSx for ONTAP to an ONTAP storage system residing on premises. Similarly, you can configure Veeam to replicate backup copies to Amazon S3 using a scale-out backup repository so that those backups are accessible to a Veeam backup server residing at the on-premises datacenter.

Failback is outside the scope of this documentation, but failback differs little from the detailed process outlined here.

Conclusion

The use case presented in this documentation focuses on proven disaster recovery technologies that highlight the integration between NetApp and VMware. NetApp ONTAP storage systems provide proven data-mirroring technologies that allow organizations to design disaster recovery solutions that span on-premises and ONTAP technologies residing with the leading cloud providers.

FSx for ONTAP on AWS is one such solution that allows for seamless integration with SnapCenter and SyncMirror for replicating application data to the cloud. Veeam Backup & Replication is another well-known technology that integrates well with NetApp ONTAP storage systems and can provide failover to vSphere-native storage.

This solution presented a disaster recovery solution using guest connect storage from an ONTAP system hosting SQL Server and Oracle application data. SnapCenter with SnapMirror provides an easy-to-manage solution for protecting application volumes on ONTAP systems and replicating them to FSx or CVO residing in the cloud. SnapCenter is a DR-enabled solution for failing over all application data to VMware Cloud on AWS.

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

- Links to solution documentation
  - NetApp Hybrid Multicloud with VMware Solutions
  - NetApp Solutions
Veeam Backup & Restore in VMware Cloud, with Amazon FSx for ONTAP

Author: Josh Powell - NetApp Solutions Engineering

Overview

Veeam Backup & Replication is an effective and reliable solution for protecting data in VMware Cloud. This solution demonstrates the proper setup and configuration for using Veeam Backup and Replication to backup and restore application VMs residing on FSx for ONTAP NFS datastores in VMware Cloud.

VMware Cloud (in AWS) supports the use of NFS datastores as supplemental storage, and FSx for NetApp ONTAP is a secure solution for customers who need to store large amounts of data for their cloud applications that can scale independent of the number of ESXi hosts in the SDDC cluster. This integrated AWS storage service offers highly efficient storage with all of the traditional NetApp ONTAP capabilities.

Use Cases

This solution addresses the following use cases:

- Backup and restore of Windows and Linux virtual machines hosted in VMC using FSx for NetApp ONTAP as a backup repository.
- Backup and restore of Microsoft SQL Server application data using FSx for NetApp ONTAP as a backup repository.
- Backup and restore of Oracle application data using FSx for NetApp ONTAP as a backup repository.

NFS Datastores Using Amazon FSx for ONTAP

All virtual machines in this solution reside on FSx for ONTAP supplemental NFS datastores. Using FSx for ONTAP as a supplemental NFS datastore has several benefits. For example, it allows you to:

- Create a scalable and highly available file system in the cloud without the need for complex setup and management.
- Integrate with your existing VMware environment, allowing you to use familiar tools and processes to manage your cloud resources.
- Benefit from the advanced data management features provided by ONTAP, such as snapshots and replication, to protect your data and ensure its availability.

Solution Deployment Overview

This list provides the high level steps necessary to configure Veeam Backup & Replication, execute backup and restore jobs using FSx for ONTAP as a backup repository, and perform restores of SQL Server and Oracle VMs and databases:

1. Create the FSx for ONTAP file system to be used as iSCSI backup repository for Veeam Backup & Replication.
2. Deploy Veeam Proxy to distribute backup workloads and mount iSCSI backup repositories hosted on FSx for ONTAP.
4. Restore SQL Server virtual machines and individual databases.
5. Restore Oracle virtual machines and individual databases.
Prerequisites

The purpose of this solution is to demonstrate data protection of virtual machines running in VMware Cloud and located on NFS Datastores hosted by FSx for NetApp ONTAP. This solution assumes the following components are configured and ready for use:

1. FSx for ONTAP filesystem with one or more NFS datastores connected to VMware Cloud.
   - vCenter server has been discovered by the Veeam Backup & Replication server using their IP address or fully qualified domain name.
3. Microsoft Windows Server VM to be installed with Veeam Backup Proxy components during the solution deployment.
4. Microsoft SQL Server VMs with VMDKs and application data residing on FSx for ONTAP NFS datastores. For this solution we had two SQL databases on two separate VMDKs.
   - Note: As a best practice database and transaction log files are placed on separate drives as this will improve performance and reliability. This is in part due to the fact that transaction logs are written sequentially, whereas database files are written randomly.
5. Oracle Database VMs with VMDKs and application data residing on FSx for ONTAP NFS datastores.
6. Linux and Windows file server VMs with VMDKs residing on FSx for ONTAP NFS datastores.
7. Veeam requires specific TCP ports for communication between servers and components in the backup environment. On Veeam backup infrastructure components, the required firewall rules are automatically created.
   For a full listing of the network port requirements refer to the Ports section of the Veeam Backup and Replication User Guide for VMware vSphere.

High Level Architecture

The testing / validation of this solution was performed in a lab that may or may not match the final deployment environment. For more information, please refer to the following sections.
Hardware / Software Components

The purpose of this solution is to demonstrate data protection of virtual machines running in VMware Cloud and located on NFS Datastores hosted by FSx for NetApp ONTAP. This solution assumes the following components are already configured and ready for use:

- Microsoft Windows VM's located on an FSx for ONTAP NFS Datastore
- Linux (CentOS) VM's located on an FSx for ONTAP NFS Datastore
- Microsoft SQL Server VM's located on an FSx for ONTAP NFS Datastore
  - Two databases hosted on separate VMDK's
- Oracle VM's located on an FSx for ONTAP NFS Datastore

Solution Deployment

In this solution we provide detailed instructions for deploying and validating a solution utilizing Veeam Backup and Replication software to perform backup and recovery of SQL Server, Oracle, and Windows and Linux file server virtual machines in a VMware Cloud SDDC on AWS. The Virtual Machines in this solution reside on a supplemental NFS datastore hosted by FSx for ONTAP. In addition, a separate FSx for ONTAP file system is used to host iSCSI volumes that will be used for Veeam backup repositories.

We will go over FSx for ONTAP file system creation, mounting iSCSI volumes to be used as backup repositories, creating and running backup jobs, and performing VM and database restores.

For detailed information on FSx for NetApp ONTAP refer to the FSx for ONTAP User Guide.

For detailed information on Veeam Backup and Replication refer to the Veeam Help Center Technical Documentation site.
For considerations and limitations when using Veeam Backup and Replication with VMware Cloud on AWS, refer to VMware Cloud on AWS and VMware Cloud on Dell EMC Support. Considerations and Limitations.

**Deploy Veeam Proxy server**

A Veeam proxy server is a component of the Veeam Backup & Replication software that acts as an intermediary between the source and the backup or replication target. The proxy server helps to optimize and accelerate data transfer during backup jobs by processing data locally and can use different Transport Modes to access data using VMware vStorage APIs for Data Protection or through direct storage access.

When choosing a Veeam proxy server design it is important to consider the number of concurrent tasks and the transport mode or type of storage access desired.

For sizing the number of proxy servers, and for their system requirements, refer to the [Veeam VMware vSphere Best Practice Guide](#).

The Veeam Data Mover is a component of the Veeam Proxy Server and utilizes a Transport Mode as a method for obtaining VM data from the source and transferring it to the target. The transport mode is specified during the configuration of the backup job. It is possible to increase the efficiency backups from NFS datastores by using direct storage access.

For more information on Transport Modes refer to the [Veeam Backup and Replication User Guide for VMware vSphere](#).

In the following step we cover deployment of the Veeam Proxy Server on a Windows VM in the VMware Cloud SDDC.
Deploy Veeam Proxy to distribute backup workloads

In this step the Veeam Proxy is deployed to an existing Windows VM. This allows backup jobs to be distributed between the primary Veeam Backup Server and the Veeam Proxy.

1. On the Veeam Backup and Replication server, open the administration console and select **Backup Infrastructure** in the lower left menu.

2. Right click on **Backup Proxies** and click on **Add VMware backup proxy**... to open the wizard.

3. In the **Add VMware Proxy** wizard click the **Add New...** button to add a new proxy server.

4. Select to add Microsoft Windows and follow the prompts to add the server:
   - Fill out the DNS name or IP address
- Select an account to use for Credentials on the new system or add new credentials
- Review the components to be installed and then click on **Apply** to begin the deployment

5. Back in the **New VMware Proxy** wizard, choose a Transport Mode. In our case we chose **Automatic Selection**.
6. Select the Connected datastores that you want the VMware Proxy to have direct access to.
7. Configure and apply any specific network traffic rules such as encryption or throttling that are desired. When complete click on the **Apply** button to complete the deployment.
Configure storage and Backup Repositories

The primary Veeam Backup server and Veeam Proxy server have access to a backup repository in the form of direct connected storage. In this section we cover creating an FSx for ONTAP file system, mounting iSCSI LUNs to the Veeam servers and creating Backup Repositories.
Create FSx for ONTAP file system

Create an FSx for ONTAP file system that will be used to host the iSCSI volumes for the Veeam Backup Repositories.

1. In the AWS console, Go to FSx and then Create file system

2. Select Amazon FSx for NetApp ONTAP and then Next to continue.

3. Fill in the file system name, deployment type, SSD storage capacity and the VPC in which the FSx for ONTAP cluster will reside. This must be a VPC configured to communicate with the virtual machine network in VMware Cloud. Click on Next.
4. Review the deployment steps and click on **Create File System** to begin the file system creation process.
Configure and mount iSCSI LUNs

Create and configure the iSCSI LUNs on FSx for ONTAP and mount to the Veeam backup and proxy servers. These LUNs will later be used to create Veeam backup repositories.

Creating an iSCSI LUN on FSx for ONTAP is a multi-step process. The first step of creating the volumes can be accomplished in the Amazon FSx Console or with the NetApp ONTAP CLI.

For more information on using FSx for ONTAP, see the FSx for ONTAP User Guide.

1. From the NetApp ONTAP CLI create the initial volumes using the following command:

   ```bash
   FSx-Backup::> volume create -vserver svm_name -volume vol_name -aggregate aggregate_name -size vol_size -type RW
   ```

2. Create LUNs using the volumes created in the previous step:

   ```bash
   FSx-Backup::> lun create -vserver svm_name -path /vol/vol_name/lun_name -size size -ostype windows -space-allocation enabled
   ```

3. Grant access to the LUNs by creating an initiator group containing the iSCSI IQN of the Veeam backup and proxy servers:

   ```bash
   FSx-Backup::> igroup create -vserver svm_name -igroup igroup_name -protocol iSCSI -ostype windows -initiator IQN
   ```

   To complete the preceding step you will need to first retrieve the IQN from the iSCSI initiator properties on the Windows servers.

4. Finally, map the LUNs to the initiator group that you just created:

   ```bash
   FSx-Backup::> lun mapping create -vserver svm_name -path /vol/vol_name/lun_name igroup igroup_name
   ```

5. To mount the iSCSI LUNs, log into the Veeam Backup & Replication Server and open iSCSI Initiator Properties. Go to the Discover tab and enter the iSCSI target IP address.
6. On the **Targets** tab, highlight the inactive LUN and click on **Connect**. Check the **Enable multi-path** box and click on **OK** to connect to the LUN.
7. In the Disk Management utility initialize the new LUN and create a volume with the desired name and drive letter. Check the **Enable multi-path** box and click on **OK** to connect to the LUN.
8. Repeat these steps to mount the iSCSI volumes on the Veeam Proxy server.
Create Veeam Backup Repositories

In the Veeam Backup and Replication console, create backup repositories for the Veeam Backup and Veeam Proxy servers. These repositories will be used as backup targets for the virtual machines backups.

1. In the Veeam Backup and Replication console click on **Backup Infrastructure** in the lower left and then select **Add Repository**

2. In the New Backup Repository wizard, enter a name for the repository and then select the server from the drop-down list and click on the **Populate** button to choose the NTFS volume that will be used.
3. On the next page choose a Mount server that will be used to mount backups to when performing advanced restores. By default this is the same server that has the repository storage connected.

4. Review your selections and click on **Apply** to start the backup repository creation.
5. Repeat these steps for any additional proxy servers.

**Configure Veeam backup jobs**

Backup jobs should be created utilizing the the Backup Repositories in the previous section. Creating backup jobs is a normal part of any storage administrator’s repertoire and we do not cover all of the steps here. For more complete information on creating backup jobs in Veeam, see the [Veeam Help Center Technical Documentation](#).

In this solution separate backup jobs were created for:

- Microsoft Windows SQL Servers
- Oracle database servers
- Windows file servers
- Linux file servers
General considerations when configuring Veeam backup jobs

1. Enable application-aware processing to create consistent backups and perform transaction log processing.

2. After enabling application-aware processing add the correct credentials with admin privileges to the application as this may be different than the guest OS credentials.

3. To manage the retention policy for the backup check the Keep certain full backups longer for archival purposes and click the Configure... button to configure the policy.
Restore Application VMs with Veeam full restore

Performing a full restore with Veeam is the first step in performing an application restore. We validated that full restores of our VMs powered on and all services were running normally.

Restoring servers is a normal part of any storage administrator’s repertoire and we do not cover all of the steps here. For more complete information on performing full restores in Veeam, see the Veeam Help Center Technical Documentation.

Restore SQL Server databases

Veeam Backup & Replication provides several options for restoring SQL Server databases. For this validation we used the Veeam Explorer for SQL Server with Instant Recovery to execute restores of our SQL Server databases. SQL Server Instant Recovery is a feature that allows you to quickly restore SQL Server databases without having to wait for a full database restore. This rapid recovery process minimizes downtime and ensures business continuity. Here’s how it works:

• Veeam Explorer mounts the backup containing the SQL Server database to be restored.

• The software publishes the database directly from the mounted files, making it accessible as a temporary database on the target SQL Server instance.

• While the temporary database is in use, Veeam Explorer redirects user queries to this database, ensuring that users can continue to access and work with the data.

• In the background, Veeam performs a full database restore, transferring data from the temporary database to the original database location.

• Once the full database restore is complete, Veeam Explorer switches user queries back to the original database and removes the temporary database.
1. In the Veeam Backup and Replication console, navigate to the list of SQL Server backups, right click on a server and select **Restore application items** and then **Microsoft SQL Server databases**.

2. In the Microsoft SQL Server Database Restore Wizard select a restore point from the list and click on **Next**.

3. Enter a **Restore reason** if desired and then, on the Summary page, click on the **Browse** button to launch Veeam Explorer for Microsoft SQL Server.
4. In Veeam Explorer expand the list of database instances, right click and select **Instant recovery** and then the specific restore point to recover to.

5. In the Instant Recovery Wizard specify the switchover type. This can either be automatically with minimal downtime, manually, or at a specified time. Then click the **Recover** button to begin the restore process.
6. The recovery process can be monitored from Veeam Explorer.

For more detailed information on performing SQL Server restore operations with Veeam Explorer refer to the Microsoft SQL Server section in the Veeam Explorers User Guide.
**Restore Oracle databases with Veeam Explorer**

Veeam Explorer for Oracle database provides the ability to perform a standard Oracle database restore or an uninterrupted restore using Instant Recovery. It also supports publishing databases for fast access, recovery of Data Guard databases and restores from RMAN backups.

For more detailed information on performing Oracle database restore operations with Veeam Explorer refer to the Oracle section in the Veeam Explorers User Guide.
In this section an Oracle database restore to a different server is covered using Veeam Explorer.

1. In the Veeam Backup and Replication console, navigate to the list of Oracle backups, right click on a server and select *Restore application items* and then *Oracle databases*.

2. In the Oracle Database Restore Wizard select a restore point from the list and click on **Next**.
3. Enter a **Restore reason** if desired and then, on the Summary page, click on the **Browse** button to launch Veeam Explorer for Oracle.
4. In Veeam Explorer expand the list of database instances, click on the database to be restored and then from the **Restore Database** drop-down menu at the top select **Restore to another server**.

5. In the Restore Wizard specify the restore point to restore from and click **Next**.
6. Specify the target server the database will be restored to and the account credentials and click Next.

7. Finally, specify the database files target location and click the Restore button to start the restore process.
Specify database files target location

Control files
/oracle/app/oradata/oradb01/control01.ctl
/oracle/app/recovery_area/oradb01/control02.ctl

Data files
/oracle/app/oradata/oradb01/system01.dbf
/oracle/app/oradata/oradb01/sysaux01.dbf
/oracle/app/oradata/oradb01/undotbs01.dbf
/oracle/app/oradata/oradb01/pdbseed/system01.dbf
/oracle/app/oradata/oradb01/pdbseed/sysaux01.dbf
/oracle/app/oradata/oradb01/users01.dbf

8. Once the database recovery is complete check that the Oracle database starts properly on the server.
Publish Oracle database to alternate server

In this section a database is published to an alternate server for fast access without launching a full restore.

1. In the Veeam Backup and Replication console, navigate to the list of Oracle backups, right click on a server and select **Restore application items** and then **Oracle databases**....

![Veeam Backup and Replication console](image)

2. In the Oracle Database Restore Wizard select a restore point from the list and click on **Next**.

![Oracle Database Restore Wizard](image)
3. Enter a **Restore reason** if desired and then, on the Summary page, click on the **Browse** button to launch Veeam Explorer for Oracle.

4. In Veeam Explorer expand the list of database instances, click on the database to be restored and then from the **Publish Database** drop-down menu at the top select **Publish to another server**.

5. In the Publish wizard, specify the restore point at which to publish the database from and click **Next**.

6. Finally, specify the target linux file system location and click on **Publish** to begin the restore process.
7. Once the publish has completed log into the target server and run the following commands to ensure the database is running:

```
oracle@ora_srv_01> sqlplus / as sysdba

SQL> select name, open_mode from v$database;
```
Conclusion

VMware Cloud is a powerful platform for running business-critical applications and storing sensitive data. A secure data protection solution is essential for businesses that rely on VMware Cloud to ensure business continuity and help protect against cyber threats and data loss. By choosing a reliable and robust data protection solution, businesses can be confident that their critical data is safe and secure, no matter what.

The use case presented in this documentation focuses on proven data protection technologies that highlight the integration between NetApp, VMware, and Veeam. FSx for ONTAP is supported as supplemental NFS datastores for VMware Cloud in AWS and is used for all virtual machine and application data. Veeam Backup & Replication is a comprehensive data protection solution designed to help businesses improve, automate, and streamline their backup and recovery processes. Veeam is used in conjunction with iSCSI backup target volumes, hosted on FSx for ONTAP, to provide a secure and easy to manage data protection solution for application data residing in VMware Cloud.

Additional Information

To learn more about the technologies presented in this solution refer to the following additional information.

- FSx for ONTAP User Guide
- Veeam Help Center Technical Documentation
- VMware Cloud on AWS Support. Considerations and Limitations

TR-4955: Disaster Recovery with FSx for ONTAP and VMC (AWS VMware Cloud)

Niyaz Mohamed, NetApp

Overview

Disaster recovery to cloud is a resilient and cost-effective way of protecting the workloads against site outages and data corruption events (for example, ransomware). With NetApp SnapMirror technology, on-premises
VMware workloads can be replicated to FSx for ONTAP running in AWS.

Disaster Recovery Orchestrator (DRO; a scripted solution with UI) can be used to seamlessly recover workloads replicated from on-premises to FSx for ONTAP. DRO automates the recovery from the SnapMirror level, through VM registration to VMC, to network mappings directly on NSX-T. This feature is included with all VMC environments.

Getting started

**Deploy and configure VMware Cloud on AWS**

VMware Cloud on AWS provides a cloud-native experience for VMware-based workloads in the AWS ecosystem. Each VMware Software-Defined Data Center (SDDC) runs in an Amazon Virtual Private Cloud (VPC) and provides a full VMware stack (including vCenter Server), NSX-T software-defined networking, vSAN software-defined storage, and one or more ESXi hosts that provide compute and storage resources to the workloads. To configure a VMC environment on AWS, follow the steps at this link. A pilot-light cluster can also be used for DR purposes.

> In the initial release, DRO supports an existing pilot-light cluster. On-demand SDDC creation will be available in an upcoming release.

**Provision and configure FSx for ONTAP**

Amazon FSx for NetApp ONTAP is a fully managed service that provides highly reliable, scalable, high-performing, and feature-rich file storage built on the popular NetApp ONTAP file system. Follow the steps at this link to provision and configure FSx for ONTAP.

**Deploy and configure SnapMirror to FSx for ONTAP**

The next step is to use NetApp BlueXP and discover the provisioned FSx for ONTAP on AWS instance and
replicate the desired datastore volumes from an on-premises environment to FSx for ONTAP with the appropriate frequency and NetApp Snapshot copy retention:

Follow the steps in this link to configure BlueXP. You can also use the NetApp ONTAP CLI to schedule replication following this link.

A SnapMirror relationship is a prerequisite and must be created beforehand.

DRO installation

To get started with DRO, use the Ubuntu operating system on a designated EC2 instance or virtual machine to make sure you meet the prerequisites. Then install the package.

Prerequisites

- Make sure that connectivity to the source and destination vCenter and storage systems exists.
- DNS resolution should be in place if you are using DNS names. Otherwise, you should use IP addresses for the vCenter and storage systems.
- Create a user with root permissions. You can also use sudo with an EC2 instance.

OS requirements

- Ubuntu 20.04 (LTS) with minimum of 2GB and 4 vCPUs
- The following packages must be installed on the designated agent VM:
  - Docker
  - Docker-compose
  - Jq

Change permissions on `docker.sock`: `sudo chmod 666 /var/run/docker.sock`. 
Install the package

1. Download the installation package on the designated virtual machine:
   
   ```
   git clone https://github.com/NetApp/DRO-AWS.git
   ```

   The agent can be installed on-premises or within an AWS VPC.

2. Unzip the package, run the deployment script, and enter the host IP (for example, 10.10.10.10).
   
   ```
   tar xvf DRO-prereq.tar
   ```

3. Navigate to the directory and run the deploy script as follows:
   
   ```
   sudo sh deploy.sh
   ```

4. Access the UI using:
   
   ```
   https://<host-ip-address>
   ```

   with the following default credentials:

   ```
   Username: admin
   Password: admin
   ```

   The password can be changed using the "Change Password" option.
DRO configuration

After FSx for ONTAP and VMC have been configured properly, you can begin configuring DRO to automate the recovery of on-premises workloads to VMC by using the read-only SnapMirror copies on FSx for ONTAP.

NetApp recommends deploying the DRO agent in AWS and also to the same VPC where FSx for ONTAP is deployed (it can be peer connected too), so that the DRO agent can communicate through the network with your on-premises components as well as with the FSx for ONTAP and VMC resources.

The first step is to discover and add the on-premises and cloud resources (both vCenter and storage) to DRO. Open DRO in a supported browser and use the default username and password (admin/admin) and Add Sites. Sites can also be added using the Discover option. Add the following platforms:

- On-premises
  - On-premises vCenter
  - ONTAP storage system
- Cloud
  - VMC vCenter
  - FSx for ONTAP
Once added, DRO performs automatic discovery and displays the VMs that have corresponding SnapMirror replicas from the source storage to FSx for ONTAP. DRO automatically detects the networks and portgroups used by the VMs and populates them.
The next step is to group the required VMs into functional groups to serve as resource groups.

**Resource groupings**

After the platforms have been added, you can group the VMs you want to recover into resource groups. DRO resource groups allow you to group a set of dependent VMs into logical groups that contain their boot orders, boot delays, and optional application validations that can be executed upon recovery.

To start creating resource groups, complete the following steps:

2. Under New resource group, select the source site from the dropdown and click Create.
3. Provide Resource Group Details and click Continue.
4. Select the appropriate VMs using the search option.
5. Select the boot order and boot delay (secs) for the selected VMs. Set the order of the power-on sequence by selecting each VM and setting up the priority for it. Three is the default value for all VMs.

Options are as follows:

1 – The first virtual machine to power on
3 – Default
5 – The last virtual machine to power on

6. Click Create Resource Group.
Replication plans

You need a plan to recover applications in the event of a disaster. Select the source and destination vCenter platforms from the drop down and pick the resource groups to be included in this plan, along with the grouping of how applications should be restored and powered on (for example, domain controllers, then tier-1, then tier-2, and so on). Such plans are sometimes also called blueprints. To define the recovery plan, navigate to the Replication Plan tab and click New Replication Plan.

To start creating a replication plan, complete the following steps:


2. Under New Replication Plan, provide a name for the plan and add recovery mappings by selecting the source site, associated vCenter, destination site, and associated vCenter.
3. After Recovery mapping is completed, select the cluster mapping.

4. Select **Resource Group Details** and click **Continue**.

5. Set the execution order for the resource group. This option enables you to select the sequence of operations when multiple resource groups exist.

6. After you are done, select the network mapping to the appropriate segment. The segments should already be provisioned within VMC, so select the appropriate segment to map the VM.

7. Based on the selection of VMs, datastore mappings are automatically selected.
SnapMirror is at the volume level. Therefore, all VMs are replicated to the replication destination. Make sure to select all VMs that are part of the datastore. If they are not selected, only the VMs that are part of the replication plan are processed.

8. Under the VM details, you can optionally resize the VM's CPU and RAM parameters; this can be very helpful when recovering large environments to smaller target clusters or for conducting DR tests without having to provision a one-to-one physical VMware infrastructure. Also, you can modify the boot order and boot delay (seconds) for all the selected VMs across the resource groups. There is an additional option to modify the boot order if there are any changes required from those selected during the resource-group boot-order selection. By default, the boot order selected during resource-group selection is used; however, any modifications can be performed at this stage.
9. Click **Create Replication Plan**.

After the replication plan is created, the failover option, the test-failover option, or the migrate option can be exercised depending on the requirements. During the failover and test-failover options, the most recent SnapMirror Snapshot copy is used, or a specific Snapshot copy can be selected from a point-in-time Snapshot copy (per the retention policy of SnapMirror). The point-in-time option can be very helpful if you are facing a corruption event like ransomware, where the most recent replicas are already compromised or encrypted. DRO shows all available points in time. To trigger failover or test failover with the configuration specified in the replication plan, you can click **Failover** or **Test failover**.
The replication plan can be monitored in the task menu:

- Use latest snapshot
- Select specific snapshot

Start Failover

After failover is triggered, the recovered items can be seen in the VMC vCenter (VMs, networks, datastores). By default, the VMs are recovered to the Workload folder.
Failback can be triggered at the replication-plan level. For a test failover, the tear-down option can be used to roll back the changes and remove the FlexClone relationship. Failback related to failover is a two-step process. Select the replication plan and select **Reverse data sync**.

Once completed, you can trigger failback to move back to original production site.
From NetApp BlueXP, we can see that replication health has broken off for the appropriate volumes (those that were mapped to VMC as read-write volumes). During test failover, DRO does not map the destination or replica volume. Instead, it makes a FlexClone copy of the required SnapMirror (or Snapshot) instance and exposes the FlexClone instance, which does not consume additional physical capacity for FSx for ONTAP. This process makes sure that the volume is not modified and replica jobs can continue even during DR tests or triage workflows. Additionally, this process makes sure that, if errors occur or corrupted data is recovered, the recovery can be cleaned up without the risk of the replica being destroyed.
Ransomware recovery

Recovering from ransomware can be a daunting task. Specifically, it can be hard for IT organizations to pinpoint where the safe point of return is and, once that is determined, to protect recovered workloads from reoccurring attacks from, for example, sleeping malware or vulnerable applications.

DRO addresses these concerns by enabling you to recover your system from any available point in time. You can also recover workloads to functional and yet isolated networks so that applications can function and communicate with each other in a location where they are not exposed to north-south traffic. This gives your security team a safe place to conduct forensics and make sure there is no hidden or sleeping malware.

Benefits

- Use of the efficient and resilient SnapMirror replication.
- Recovery to any available point in time with Snapshot copy retention.
- Full automation of all required steps to recover hundreds to thousands of VMs from the storage, compute, network, and application validation steps.
- Workload recovery with ONTAP FlexClone technology using a method that doesn’t change the replicated volume.
  - Avoids risk of data corruption for volumes or Snapshot copies.
  - Avoids replication interruptions during DR test workflows.
  - Potential use of DR data with cloud computing resources for workflows beyond DR such as DevTest, security testing, patch or upgrade testing, and remediation testing.
- CPU and RAM optimization to help lower cloud costs by allowing recovery to smaller compute clusters.

Using Veeam Replication and FSx for ONTAP for Disaster recovery to VMware Cloud on AWS

Author: Niyaz Mohamed - NetApp Solutions Engineering
Overview

Amazon FSx for NetApp ONTAP integration with VMware Cloud on AWS is an AWS-managed external NFS datastore built on NetApp’s ONTAP file system that can be attached to a cluster in the SDDC. It provides customers with flexible, high-performance virtualized storage infrastructure that scales independently of compute resources.

For those customers looking to use VMware Cloud on AWS SDDC as the disaster recovery target, FSx for ONTAP datastores can be used to replicate data from on-premises using any validated third-party solution that provides VM replication capability. By adding FSx for ONTAP datastore, it will enable cost optimised deployment than building VMware cloud on AWS SDDC with enormous amount of ESXi hosts just to accommodate the storage.

This approach also helps customers to use pilot light cluster in VMC along with FSx for ONTAP datastores to host the VM replicas. The same process can also be extended as a migration option to VMware Cloud on AWS by gracefully failing over the replication plan.

Problem Statement

This document describes how to use FSx for ONTAP datastore and Veeam Backup and replication to set up disaster recovery for on-premises VMware VMs to VMware Cloud on AWS using the VM replication functionality.

Veeam Backup & Replication allows onsite and remote replication for disaster recovery (DR). When virtual machines are replicated, Veeam Backup & Replication creates an exact copy of the VMs in the native VMware vSphere format on the target VMware Cloud on AWS SDDC cluster and keeps the copy synchronized with the original VM.

Replication provides the best recovery time objective (RTO) values as there is a copy of a VM in the ready-to-start state. This replication mechanism ensures that the workloads can quickly start in VMware Cloud on AWS SDDC in case of a disaster event. The Veeam Backup & Replication software also optimizes traffic transmission for replication over WAN and slow connections. In addition, it also filters out duplicate data blocks, zero data blocks, swap files and excluded VM guest OS files, and compresses the replica traffic.

To prevent replication jobs from consuming the entire network bandwidth, WAN accelerators and network throttling rules can be put in place. The replication process in Veeam Backup & Replication is job driven which means replication is performed by configuring replication jobs. In case of a disaster event, failover can be triggered to recover the VMs by failing over to its replica copy.

When failover is performed, a replicated VM takes over the role of the original VM. Fail over can be performed to the latest state of a replica or to any of its good known restore points. This enables ransomware recovery or isolated testing as needed. In Veeam Backup & Replication, failover and failback are temporary intermediate step that should be further finalized. Veeam Backup & Replication offers multiple options to handle different disaster recovery scenarios.

Solution Deployment

High level steps

1. Veeam Backup and Replication software is running in on-premises environment with appropriate network connectivity.

2. Configure VMware Cloud on AWS, see the VMware Cloud Tech Zone article VMware Cloud on AWS integration with Amazon FSx for NetApp ONTAP Deployment Guide to deploy, configure VMware Cloud on
AWS SDDC and FSx for ONTAP as NFS datastore. (A pilot-light environment set up with a minimal configuration can be used for DR purposes. VMs will fail over to this cluster in the event of an incident, and additional nodes can be added).

3. Set up replication jobs to create VM replicas using Veeam Backup and Replication.
4. Create failover plan and perform failover.
5. Switch back to production VMs once the disaster event is complete and primary site is Up.

**Pre-requisites for Veeam VM Replication to VMC and FSx for ONTAP datastores**

1. Ensure Veeam Backup & Replication backup VM is connected to the source vCenter as well as the target VMware cloud on AWS SDDC clusters.
2. The backup server must be able to resolve short names and connect to source and target vCenters.
3. The target FSx for ONTAP datastore must have enough free space to store VMDKs of replicated VMs

For additional information, refer to "Considerations and Limitations" covered here.

**Deployment Details**

1995
Step 1: Replicate VMs

Veeam Backup & Replication leverages VMware vSphere snapshot capabilities and during replication, Veeam Backup & Replication requests VMware vSphere to create a VM snapshot. The VM snapshot is the point-in-time copy of a VM that includes virtual disks, system state, configuration and so on. Veeam Backup & Replication uses the snapshot as a source of data for replication.

To replicate VMs, follow the below steps:

1. Open the Veeam Backup & Replication Console.
2. On the Home view, select Replication Job > Virtual machine > VMware vSphere.
3. Specify a job name and select the appropriate advanced control checkbox. Click Next.
   - Select the Replica seeding check box if connectivity between on-premises and AWS has restricted bandwidth.
   - Select the Network remapping (for AWS VMC sites with different networks) check box if segments on VMware Cloud on AWS SDDC do not match that of on-premises site networks.
   - If the IP addressing scheme in on-premises production site differs from the scheme in the AWS VMC site, select the Replica re-IP (for DR sites with different IP addressing scheme) check box.

4. Select the VMs that needs to be replicated to FSx for ONTAP datastore attached to VMware Cloud on AWS SDDC in the Virtual Machines step. The Virtual machines can be placed on vSAN to fill the available vSAN datastore capacity. In a pilot light cluster, the usable capacity of a 3-node cluster will be limited. The rest of the data can be replicated to FSx for ONTAP datastores. Click Add, then in the Add Object window select the necessary VMs or VM containers and click Add. Click Next.

5. After that, select the destination as VMware Cloud on AWS SDDC cluster / host and the appropriate resource pool, VM folder and FSx for ONTAP datastore for VM replicas. Then Click Next.

6. In the next step, create the mapping between source and destination virtual network as needed.

7. In the Job Settings step, specify the backup repository that will store metadata for VM replicas, retention policy and so on.

8. Update the Source and Target proxy servers in the Data Transfer step and leave Automatic selection (default) and keep Direct option selected and click Next.

9. At the Guest Processing step, select Enable application-aware processing option as needed. Click Next.

10. Choose the replication schedule to run the replication job to run on a regular basis.

11. At the Summary step of the wizard, review details of the replication job. To start the job right after the wizard is closed, select the Run the job when I click Finish check box, otherwise leave the check box unselected. Then click Finish to close the wizard.
Once the replication job starts, the VMs with the suffix specified will be populated on the destination VMC SDDC cluster / host.

For additional information for Veeam replication, refer to How Replication Works.

**Step 2: Create a failover plan**

When the initial replication or seeding is complete, create the failover plan. Failover plan helps in performing failover for dependent VMs one by one or as a group automatically. Failover plan is the blueprint for the order in which the VMs are processed including the boot delays. The failover plan also helps to ensure that critical dependant VMs are already running.

To create the plan, navigate to the new sub section called Replicas and select Failover Plan. Choose the appropriate VMs. Veeam Backup & Replication will look for the closest restore points to this point in time and use them to start VM replicas.

- The failover plan can only be added once the initial replication is complete and the VM replicas are in Ready state.
- The maximum number of VMs that can be started simultaneously when running a failover plan is 10.
- During the failover process, the source VMs will not be powered off.

To create the **Failover Plan**, do the following:

1. On the Home view, select **Failover Plan > VMware vSphere**.
2. Next, provide a name and a description to the plan. Pre and Post-failover script can be added as required. For instance, run a script to shutdown VMs before starting the replicated VMs.
3. Add the VMs to the plan and modify the VM boot order and boot delays to meet the application dependencies.

For additional information for creating replication jobs, refer to Creating Replication Jobs.
Step 3: Run the failover plan

During failover, the source VM in the production site is switched over to its replica at the disaster recovery site. As part of the failover process, Veeam Backup & Replication restores the VM replica to the required restore point and moves all I/O activities from the source VM to its replica. Replicas can be used not only in case of a disaster, but also to simulate DR drills. During failover simulation, the source VM remains running. Once all the necessary tests have been conducted, you can undo the failover and return to normal operations.

Make sure network segmentation is in place to avoid IP conflicts during DR drills.

To start the failover plan, simply click in **Failover Plans** tab and right click on the failover plan. Select **Start**. This will failover using the latest restore points of VM replicas. To fail over to specific restore points of VM replicas, select **Start to**.

1. Veeam Backup & Replication stops all replication activities for the source VM until its replica is returned to the Ready state.

For detailed information about failover plans, refer to **Failover Plans**.
Step 4: Failback to the Production site

When the failover plan is running, it is considered as an intermediate step and needs to be finalized based on the requirement. The options include the following:

- **Failback to production** - switch back to the original VM and transfer all changes that took place while the VM replica was running to the original VM.

  When you perform failback, changes are only transferred but not published. Choose **Commit failback** (once the original VM is confirmed to work as expected) or **Undo failback** to get back to the VM replica if the original VM is not working as expected.

- **Undo failover** - switch back to the original VM and discard all changes made to the VM replica while it was running.

- **Permanent Failover** - permanently switch from the original VM to a VM replica and use this replica as the original VM.

In this demo, Failback to production was chosen. Failback to the original VM was selected during the Destination step of the wizard and “Power on VM after restoring” check box was enabled.

Failback commit is one of the ways to finalize failback operation. When failback is committed, it confirms that the changes sent to the VM which is failed back (the production VM) are working as expected. After the commit operation, Veeam Backup & Replication resumes replication activities for the production VM.

For detailed information about the failback process, refer Veeam documentation for **Failover and Failback for replication**.

After failback to production is successful, the VMs are all restored back to the original production site.

Conclusion

FSx for ONTAP datastore capability enables Veeam or any validated third-party tool to provide low-cost DR solution using Pilot light cluster and without standing up large number of hosts in the cluster just to accommodate the VM replica copy. This provides a powerful solution to handle a tailored, customized disaster recovery plan and also allows to reuse existing backup products in house to meet the DR needs, thus enabling cloud-based disaster recovery by exiting DR datacentres on-premises. Failover can be done as planned failover or failover with a click of a button when disaster occurs, and decision is made to activate the DR site.

To learn more about this process, feel free to follow the detailed walkthrough video.

[Video Link](https://netapp.hosted.panopto.com/Panopto/Pages/Embed.aspx?id=15fed205-8614-4ef7-b2d0-b061015e925a)
Migrating Workloads on AWS / VMC

TR 4942: Migrate Workloads to FSx ONTAP datastore using VMware HCX

Author(s): NetApp Solutions Engineering

Overview: Migrating virtual machines with VMware HCX, FSx ONTAP supplemental datastores, and VMware Cloud

A common use case for VMware Cloud (VMC) on Amazon Web Services (AWS), with its supplemental NFS datastore on Amazon FSx for NetApp ONTAP, is the migration of VMware workloads. VMware HCX is a preferred option and provides various migration methods to move on-premises virtual machines (VMs) and their data, running on any VMware supported datastores, to VMC datastores, which includes supplemental NFS datastores on FSx for ONTAP.

VMware HCX is primarily a mobility platform that is designed to simplify workload migration, workload rebalancing, and business continuity across clouds. It is included as part of VMware Cloud on AWS and offers many ways to migrate workloads and can be used for disaster recovery (DR) operations.

This document provides step-by-step guidance for deploying and configuring VMware HCX, including all its main components, on-premises and on the cloud data center side, which enables various VM migration mechanisms.

For more information, see Introduction to HCX Deployments and Install Checklist B - HCX with a VMware Cloud on AWS SDDC Destination Environment.

High-level steps

This list provides the high-level steps to install and configure VMware HCX:

1. Activate HCX for the VMC software-defined data center (SDDC) through VMware Cloud Services Console.
2. Download and deploy the HCX Connector OVA installer in the on-premises vCenter Server.
3. Activate HCX with a license key.
4. Pair on-premises VMware HCX Connector with VMC HCX Cloud Manager.
5. Configure the network profile, compute profile, and service mesh.
6. (Optional) Perform Network Extension to extend the network and avoid re-IP.
7. Validate the appliance status and ensure that migration is possible.
8. Migrate the VM workloads.
Prerequisites

Before you begin, make sure the following prerequisites are met. For more information, see Preparing for HCX Installation. After the prerequisites are in place, including connectivity, configure and activate HCX by generating a license key from the VMware HCX Console at VMC. After HCX is activated, the vCenter Plug-in is deployed and can be accessed by using the vCenter Console for management.

The following installation steps must be completed before proceeding with HCX activation and deployment:

1. Use an existing VMC SDDC or create a new SDDC following this NetApp link or this VMware link.
2. The network path from the on-premises vCenter environment to the VMC SDDC must support migration of VMs by using vMotion.
3. Make sure the required firewall rules and ports are allowed for vMotion traffic between the on-premises vCenter Server and the SDDC vCenter.
4. The FSx for ONTAP NFS volume should be mounted as a supplemental datastore in the VMC SDDC. To attach the NFS datastores to the appropriate cluster, follow the steps outlined in this NetApp link or this VMware link.

High Level Architecture

For testing purposes, the on-premises lab environment used for this validation was connected through a site-to-site VPN to AWS VPC, which allowed on-premises connectivity to AWS and to VMware cloud SDDC through External transit gateway. HCX migration and network extension traffic flows over the internet between on-premises and VMware cloud destination SDDC. This architecture can be modified to use Direct Connect private virtual interfaces.

The following image depicts the high-level architecture.
Solution Deployment

Follow the series of steps to complete the deployment of this solution:

Step 1: Activate HCX through VMC SDDC using the Add-ons option

To perform the installation, complete the following steps:

1. Log in to the VMC Console at vmc.vmware.com and access Inventory.
2. To select the appropriate SDDC and access Add-ons, click View Details on SDDC and select the Add Ons tab.
3. Click Activate for VMware HCX.
   
   This step takes up to 25 minutes to complete.

4. After the deployment is complete, validate the deployment by confirming that HCX Manager and its associated plug-ins are available in vCenter Console.

5. Create the appropriate Management Gateway firewalls to open the ports necessary to access HCX Cloud Manager. HCX Cloud Manager is now ready for HCX operations.
Step 2: Deploy the installer OVA in the on-premises vCenter Server

For the on-premises Connector to communicate with the HCX Manager in VMC, make sure that the appropriate firewall ports are open in the on-premises environment.

1. From the VMC Console, navigate to the HCX Dashboard, go to Administration, and select the Systems Update tab. Click Request a Download Link for the HCX Connector OVA image.

2. With the HCX Connector downloaded, deploy the OVA in the on-premises vCenter Server. Right-click vSphere Cluster and select the Deploy OVF Template option.

3. Enter the required information in the Deploy OVF Template wizard, click Next and then Finish to deploy the VMware HCX Connector OVA.

Step 3: Activate HCX Connector with the license key

After you deploy the VMware HCX Connector OVA on-premises and start the appliance, complete the following steps to activate HCX Connector. Generate the license key from the VMware HCX Console at VMC and input the license during the VMware HCX Connector setup.

1. From the VMware Cloud Console, go to Inventory, select the SDDC, and click View Details. From the Add Ons tab, in the VMware HCX tile, click Open HCX.

2. From the Activation Keys tab, click Create Activation Key. Select the System Type as HCX Connector and click Confirm to generate the key. Copy the activation key.

A separate key is required for each HCX Connector deployed on-premises.


Use the password defined during the OVA deployment.

4. In the Licensing section, enter the activation key copied from step 2 and click Activate.

The on-premises HCX Connector must have internet access for the activation to complete successfully.

5. Under Datacenter Location, provide the desired location for installing the VMware HCX Manager on-premises. Click Continue.

6. Under System Name, update the name and click Continue.

7. Select Yes and then Continue.

8. Under Connect Your vCenter, provide the IP address or fully qualified domain name (FQDN) and the credentials for the vCenter Server and click Continue.

Use the FQDN to avoid communication issues later.

9. Under Configure SSO/PSC, provide the Platform Services Controller’s FQDN or IP address and click Continue.

Enter the vCenter Server’s IP address or FQDN.

10. Verify that the information is entered correctly and click Restart.

11. After complete, the vCenter Server is displayed as green. Both the vCenter Server and SSO must
have the correct configuration parameters, which should be the same as the previous page.

This process should take approximately 10–20 minutes and for the plug-in to be added to the vCenter Server.
Step 4: Pair on-premises VMware HCX Connector with VMC HCX Cloud Manager

1. To create a site pair between the on-premises vCenter Server and the VMC SDDC, log in to the on-premises vCenter Server and access the HCX vSphere Web Client Plug-in.

   ![vSphere Client Screenshot]

   Shortcuts
   
   Infinities
   
   Monitoring
   
   Task Console
   
   Event Console
   
   Monitoring
   
   VM Storage Policies
   
   Host Profiles
   
   Lifecycle Manager
   
   ONTAP tools

   Administration

   Licensing

2. Under Infrastructure, click Add a Site Pairing. To authenticate the remote site, enter the VMC HCX Cloud Manager URL or IP address and the credentials for the CloudAdmin role.

   ![vSphere Client Screenshot: Site Pairing]

   HCX information can be retrieved from the SDDC Settings page.
3. To initiate the site pairing, click Connect.

![Connect to Remote Site](image)

- VMware HCX Connector must be able to communicate with the HCX Cloud Manager IP over port 443.

4. After the pairing is created, the newly configured site pairing is available on the HCX Dashboard.
Step 5: Configure the network profile, compute profile, and service mesh

The VMware HCX Interconnect (HCX-IX) appliance provides secure tunnel capabilities over the internet and private connections to the target site that enable replication and vMotion-based capabilities. The interconnect provides encryption, traffic engineering, and an SD-WAN. To create the HCI-IX Interconnect Appliance, complete the following steps:

1. Under Infrastructure, select Interconnect > Multi-Site Service Mesh > Compute Profiles > Create Compute Profile.

   Compute profiles contain the compute, storage, and network deployment parameters required to deploy an interconnect virtual appliance. They also specify which portion of the VMware data center will be accessible to the HCX service.

   For detailed instructions, see Creating a Compute Profile.

2. After the compute profile is created, create the network profile by selecting Multi-Site Service Mesh > Network Profiles > Create Network Profile.

3. The network profile defines a range of IP address and networks that will be used by HCX for its virtual appliances.

   This will require two or more IP address. These IP addresses will be assigned from the management network to virtual appliances.
For detailed instructions, see [Creating a Network Profile](#).

If you are connecting with an SD-WAN over the internet, you have to reserve public IPs under the Networking and Security section.

4. To create a service mesh, select the Service Mesh tab within the Interconnect option and select on-premises and VMC SDDC sites.

The service mesh establishes a local and remote compute and network profile pair.

Part of this process involves deploying HCX appliances that will be automatically configured on both the source and target sites, creating a secure transport fabric.

5. Select the source and remote compute profiles and click Continue.
6. Select the service to be activated and click Continue.

7. Create a name for the service mesh and click Finish to begin the creation process. The deployment should take approximately 30 minutes to complete. After the service mesh is configured, the virtual infrastructure and networking required to migrate the workload VMs has been created.

**Info**

An HCX Enterprise license is required for Replication Assisted vMotion Migration, SRM Integration, and OS Assisted Migration.
Step 6: Migrating Workloads

HCX provides bidirectional migration services between two or more distinct environments such as on-premises and VMC SDDCs. Application workloads can be migrated to and from HCX activated sites using a variety of migration technologies such as HCX bulk migration, HCX vMotion, HCX Cold migration, HCX Replication Assisted vMotion (available with HCX Enterprise edition), and HCX OS Assisted Migration (available with HCX Enterprise edition).

To learn more about available HCX migration technologies, see VMware HCX Migration Types

The HCX-IX appliance uses the Mobility Agent service to perform vMotion, Cold, and Replication Assisted vMotion (RAV) migrations.

The HCX-IX appliance adds the Mobility Agent service as a host object in the vCenter Server. The processor, memory, storage and networking resources displayed on this object do not represent actual consumption on the physical hypervisor hosting the IX appliance.
This section describes the HCX vMotion mechanism. This migration technology uses the VMware vMotion protocol to migrate a VM to VMC SDDC. The vMotion migration option is used for migrating the VM state of a single VM at a time. There is no service interruption during this migration method.

- Network Extension should be in place (for the port group in which the VM is attached) in order to migrate the VM without the need to make an IP address change.

1. From the on-premises vSphere client, go to Inventory, right-click on the VM to be migrated, and select HCX Actions > Migrate to HCX Target Site.

2. In the Migrate Virtual Machine wizard, select the Remote Site Connection (target VMC SDDC).
3. Add a group name and under Transfer and Placement, update the mandatory fields (Cluster, Storage, and Destination Network), Click Validate.

4. After the validation checks are complete, click Go to initiate the migration.
The vMotion transfer captures the VM active memory, its execution state, its IP address, and its MAC address. For more information about the requirements and limitations of HCX vMotion, see Understanding VMware HCX vMotion and Cold Migration.

5. You can monitor the progress and completion of the vMotion from the HCX > Migration dashboard.
As you might have noticed from VMware documentation, VMware HCX Replication Assisted vMotion (RAV) combines the benefits of bulk migration and vMotion. Bulk migration uses vSphere Replication to migrate multiple VMs in parallel—the VM gets rebooted during switchover. HCX vMotion migrates with no downtime, but it is performed serially one VM at a time in a replication group. RAV replicates the VM in parallel and keeps it in sync until the switchover window. During the switchover process, it migrates one VM at a time with no downtime for the VM.

The following screenshot show the migration profile as Replication Assisted vMotion.

The duration of the replication might be longer compared to the vMotion of a small number of VMs. With RAV, only sync the deltas and include the memory contents. The following is a screenshot of the migration status—it shows how the start time of the migration is the same and the end time is different for each VM.

For additional information about the HCX migration options and on how to migrate workloads from on-premises to VMware Cloud on AWS using HCX, see the VMware HCX User Guide.
VMware HCX vMotion requires 100Mbps or higher throughput capability.

The target VMC FSx for ONTAP datastore must have sufficient space to accommodate the migration.

Conclusion

Whether you are targeting all-cloud or hybrid cloud and data residing on any type/vendor storage in on-premises, Amazon FSx for NetApp ONTAP along with HCX provide excellent options to deploy and migrate the workloads while reducing the TCO by making the data requirements seamless to the application layer. Whatever the use case, choose VMC along with FSx for ONTAP datastore for rapid realization of cloud benefits, consistent infrastructure, and operations across on-premises and multiple clouds, bidirectional portability of workloads, and enterprise-grade capacity and performance. It is the same familiar process and procedures used to connect the storage and migrate VMs using VMware vSphere replication, VMware vMotion or even NFC copy.

Takeaways

The key points of this document include:

- You can now use Amazon FSx ONTAP as a datastore with VMC SDDC.
- You can easily migrate data from any on-premises datacenter to VMC running with FSx for ONTAP datastore
- You can easily grow and shrink the FSx ONTAP datastore to meet the capacity and performance requirements during migration activity.

Where to find additional information

To learn more about the information described in this document, refer to the following website links:

- VMware Cloud documentation
  https://docs.vmware.com/en/VMware-Cloud-on-AWS/
- Amazon FSx for NetApp ONTAP documentation
  https://docs.aws.amazon.com/fsx/latest/ONTAPGuide
- VMware HCX User Guide

Region Availability – Supplemental NFS datastore for VMC

Unresolved directive in ehc/aws-regions.adoc - include::/_include/aws-region-support.adoc[]
NetApp Hybrid Multicloud with VMware Solutions

Protecting Workloads on Azure / AVS

Disaster Recovery with ANF and JetStream

Disaster recovery to cloud is a resilient and cost-effective way of protecting the workloads against site outages and data corruption events (for example, ransomware). Using the VMware VAIO framework, on-premises VMware workloads can be replicated to Azure Blob storage and recovered, enabling minimal or close to no data loss and near-zero RTO.

JetStream DR can be used to seamlessly recover the workloads replicated from on-premises to AVS and specifically to Azure NetApp Files. It enables cost-effective disaster recovery by using minimal resources at the DR site and cost-effective cloud storage. JetStream DR automates recovery to ANF datastores via Azure Blob Storage. JetStream DR recovers independent VMs or groups of related VMs into recovery site infrastructure according to network mapping and provides point-in-time recovery for ransomware protection.

This document provides an understanding of the JetStream DR principles of operations and its main components.
Solution deployment overview

1. Install JetStream DR software in the on-premises data center.
   a. Download the JetStream DR software bundle from Azure Marketplace (ZIP) and deploy the JetStream DR MSA (OVA) in the designated cluster.
   b. Configure the cluster with the I/O filter package (install JetStream VIB).
   c. Provision Azure Blob (Azure Storage Account) in the same region as the DR AVS cluster.
   d. Deploy DRVA appliances and assign replication log volumes (VMDK from existing datastore or shared iSCSI storage).
   e. Create protected domains (groups of related VMs) and assign DRVAs and Azure Blob Storage/ANF.
   f. Start protection.

2. Install JetStream DR software in the Azure VMware Solution private cloud.
   a. Use the Run command to install and configure JetStream DR.
   b. Add the same Azure Blob container and discover domains using the Scan Domains option.
   c. Deploy required DRVA appliances.
   d. Create replication log volumes using available vSAN or ANF datastores.
   e. Import protected domains and configure RocVA (recovery VA) to use ANF datastore for VM placements.
   f. Select the appropriate failover option and start continuous rehydration for near-zero RTO domains or VMs.

3. During a disaster event, trigger failover to Azure NetApp Files datastores in the designated AVS DR site.

4. Invoke failback to the protected site after the protected site has been recovered. Before starting, make sure that the prerequisites are met as indicated in this link and also run the Bandwidth Testing Tool (BWT) provided by JetStream Software to evaluate the potential performance of Azure Blob storage and its replication bandwidth when used with JetStream DR software. After the pre-requisites, including connectivity, are in place, set up and subscribe to JetStream DR for AVS from the Azure Marketplace. After the software bundle is downloaded, proceed with the installation process described above.

When planning and starting protection for a large number of VMs (for example, 100+), use the Capacity Planning Tool (CPT) from the JetStream DR Automation Toolkit. Provide a list of VMs to be protected together with their RTO and recovery group preferences, and then run CPT.

CPT performs the following functions:

- Combining VMs into protection domains according to their RTO.
- Defining the optimal number of DRVAs and their resources.
- Estimating required replication bandwidth.
- Identifying replication log volume characteristics (capacity, bandwidth, and so on).
- Estimating required object storage capacity, and more.
The number and content of domains prescribed depend upon various VM characteristics such as average IOPS, total capacity, priority (which defines failover order), RTO, and others.

**Install JetStream DR in On-Premises Datacenter**

JetStream DR software consists of three major components: JetStream DR Management Server Virtual Appliance (MSA), DR Virtual Appliance (DRVA), and host components (I/O Filter packages). MSA is used to install and configure host components on the compute cluster and then to administer JetStream DR software. The following list provides a high-level description of the installation process:
How to install JetStream DR for on-premises

1. Check prerequisites.
2. Run the Capacity Planning Tool for resource and configuration recommendations (optional but recommended for proof-of-concept trials).
3. Deploy the JetStream DR MSA to a vSphere host in the designated cluster.
4. Launch the MSA using its DNS name in a browser.
5. Register the vCenter server with the MSA. To perform the installation, complete the following detailed steps:
6. After JetStream DR MSA has been deployed and the vCenter Server has been registered, access the JetStream DR plug-in using the vSphere Web Client. This can be done by navigating to Datacenter > Configure > JetStream DR.

7. From the JetStream DR interface, select the appropriate cluster.

8. Configure the cluster with the I/O filter package.
9. Add Azure Blob Storage located at the recovery site.

10. Deploy a DR Virtual Appliance (DRVA) from the Appliances tab.

DRVAs can be automatically created by CPT, but for POC trials we recommend configuring and running the DR cycle manually (start protection > failover > failback).

The JetStream DRVA is a virtual appliance that facilitates key functions in the data replication process. A protected cluster must contain at least one DRVA, and typically one DRVA is configured per host. Each DRVA can manage multiple protected domains.

In this example, four DRVA’s were created for 80 virtual machines.

1. Create replication log volumes for each DRVA using VMDK from the datastores available or independent shared iSCSI storage pools.

2. From the Protected Domains tab, create the required number of protected domains using information
about the Azure Blob Storage site, DRVA instance, and replication log. A protected domain defines a specific VM or set of VMs within the cluster that are protected together and assigned a priority order for failover/failback operations.

3. Select VMs you want to protect and start VM protection of the protected domain. This begins data replication to the designated Blob Store.

- Verify that the same protection mode is used for all VMs in a protected domain.
- Write-Back (VMDK) mode can offer higher performance.

Verify that replication log volumes are placed on high performance storage.

- Failover run books can be configured to group the VMs (called Recovery Group), set boot order sequence, and modify the CPU/memory settings along with IP configurations.
Install JetStream DR for AVS in an Azure VMware Solution private cloud using the Run command

A best practice for a recovery site (AVS) is to create a three-node pilot-light cluster in advance. This allows the recovery site infrastructure to be preconfigured, including the following items:

- Destination networking segments, firewalls, services like DHCP and DNS, and so on.
- Installation of JetStream DR for AVS
- Configuration of ANF volumes as datastores, and moreJetStream DR supports near-zero RTO mode for mission-critical domains. For these domains, destination storage should be preinstalled. ANF is a recommended storage type in this case.

Network configuration including segment creation should be configured on the AVS cluster to match on-premises requirements.

Depending on the SLA and RTO requirements, continuous failover or regular (standard) failover mode can be used. For near-zero RTO, continuous rehydration should be started at the recovery site.
How to install JetStream DR for AVS in a private cloud

To install JetStream DR for AVS on an Azure VMware Solution private cloud, complete the following steps:

1. From the Azure portal, go to the Azure VMware solution, select the private cloud, and select Run command > Packages > JSDR.Configuration.

   The default CloudAdmin user in Azure VMware Solution doesn’t have sufficient privileges to install JetStream DR for AVS. Azure VMware Solution enables simplified and automated installation of JetStream DR by invoking the Azure VMware Solution Run command for JetStream DR.

   The following screenshot shows installation using a DHCP-based IP address.

2. After JetStream DR for AVS installation is complete, refresh the browser. To access the JetStream DR UI, go to SDDC Datacenter > Configure > JetStream DR.
3. From the JetStream DR interface, add the Azure Blob Storage account that was used to protect the on-premises cluster as a storage site and then run the Scan Domains option.

4. After the protected domains are imported, deploy DRVA appliances. In this example, continuous rehydration is started manually from the recovery site using the JetStream DR UI.

   These steps can also be automated using CPT created plans.

5. Create replication log volumes using available vSAN or ANF datastores.

6. Import the protected domains and configure the Recovery VA to use the ANF datastore for VM placements.
Make sure that DHCP is enabled on the selected segment and enough IPs are available. Dynamic IPs are temporarily used while domains are recovering. Each recovering VM (including continuous rehydration) requires an individual dynamic IP. After recovery is complete, the IP is released and can be reused.

7. Select the appropriate failover option (continuous failover or failover). In this example, continuous rehydration (continuous failover) is selected.
How to perform a Failover / Failback

1. After a disaster occurs in the protected cluster of the on-premises environment (partial or full failure), trigger the failover.

   CPT can be used to execute the failover plan to recover the VMs from Azure Blob Storage into the AVS cluster recovery site.

   After failover (for continuous or standard rehydration) when the protected VMs have been started in AVS, protection is automatically resumed and JetStream DR continues to replicate their data into the appropriate/original containers in Azure Blob Storage.

   The task bar shows progress of failover activities.
2. When the task is complete, access the recovered VMs and business continues as normal.

After the primary site is up and running again, failback can be performed. VM protection is resumed and data consistency should be checked.

3. Restore the on-premises environment. Depending upon the type of disaster incident, it might be necessary to restore and/or verify the configuration of the protected cluster. If necessary, JetStream DR software might need to be reinstalled.

Note: The `recovery_utility_prepare_failback` script provided in the Automation Toolkit can be used to help clean the original protected site of any obsolete VMs, domain information, and so on.

4. Access the restored on-premises environment, go to the Jetstream DR UI, and select the appropriate protected domain. After the protected site is ready for failback, select the Failback option in the UI.
The CPT generated failback plan can also be used to initiate the return of the VMs and their data from the object store back to the original VMware environment.

Specify the maximum delay after pausing VMs in the recovery site and restarting in the protected site. This time includes completing replication after stopping failover VMs, the time to clean recovery site, and the time to recreate VMs in protected site. The NetApp recommended value is 10 minutes.

Complete the failback process, and then confirm the resumption of VM protection and data consistency.

Ransomeware Recovery

Recovering from ransomware can be a daunting task. Specifically, it can be hard for IT organizations to determine the safe point of return and, once determined, how to ensure that recovered workloads are safeguarded from the attacks reoccurring (from sleeping malware or through vulnerable applications).

JetStream DR for AVS together with Azure NetApp Files datastores can address these concerns by allowing organizations to recover from available points in time, so that workloads are recovered to a functional, isolated network if required. Recovery allows applications to function and communicate with each other while not exposing them to north-south traffic, thereby giving security teams a safe place to perform forensics and other necessary remediation.

Disaster Recovery with CVO and AVS (guest-connected storage)

Overview

Authors: Ravi BCB and Niyaz Mohamed, NetApp

Disaster recovery to cloud is a resilient and cost-effective way of protecting workloads against site outages and data corruption events such as ransomware. With NetApp SnapMirror, on-premises VMware workloads that use guest-connected storage can be replicated to NetApp Cloud Volumes ONTAP running in Azure. This covers application data; however, what about the actual VMs themselves. Disaster recovery should cover all dependent components, including virtual machines, VMDKs, application data, and more. To accomplish this,
SnapMirror along with JetStream can be used to seamlessly recover workloads replicated from on-premises to Cloud Volumes ONTAP while using vSAN storage for VM VMDKs.

This document provides a step-by-step approach for setting up and performing disaster recovery that uses NetApp SnapMirror, JetStream, and the Azure VMware Solution (AVS).

**Assumptions**

This document focuses on in-guest storage for application data (also known as guest connected), and we assume that the on-premises environment is using SnapCenter for application-consistent backups.

This document applies to any third-party backup or recovery solution. Depending on the solution used in the environment, follow best practices to create backup policies that meet organizational SLAs.

For connectivity between the on-premises environment and the Azure virtual network, use the express route global reach or a virtual WAN with a VPN gateway. Segments should be created based on the on-premises vLAN design.

There are multiple options for connecting on-premises datacenters to Azure, which prevents us from outlining a specific workflow in this document. Refer to the Azure documentation for the appropriate on-premises-to-Azure connectivity method.

**Deploying the DR Solution**

**Solution Deployment Overview**

1. Make sure that application data is backed up using SnapCenter with the necessary RPO requirements.
2. Provision Cloud Volumes ONTAP with the correct instance size using Cloud manager within the
appropriate subscription and virtual network.

a. Configure SnapMirror for the relevant application volumes.

b. Update the backup policies in SnapCenter to trigger SnapMirror updates after the scheduled jobs.

3. Install the JetStream DR software in the on-premises data center and start protection for virtual machines.

4. Install JetStream DR software in the Azure VMware Solution private cloud.

5. During a disaster event, break the SnapMirror relationship using Cloud Manager and trigger failover of virtual machines to Azure NetApp Files or to vSAN datastores in the designated AVS DR site.

   a. Reconnect the iSCSI LUNs and NFS mounts for the application VMs.

6. Invoke failback to the protected site by reverse resyncing SnapMirror after the primary site has been recovered.

### Deployment Details

#### Configure CVO on Azure and replicate volumes to CVO

The first step is to configure Cloud Volumes ONTAP on Azure ([Link](#)) and replicate the desired volumes to Cloud Volumes ONTAP with the desired frequencies and snapshot retentions.

<table>
<thead>
<tr>
<th>Health Status</th>
<th>Source Volume</th>
<th>Target Volume</th>
<th>Total Transfer Time</th>
<th>Status</th>
<th>Mirror State</th>
<th>Last Successful Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>gcsdrsql/db-sc46 ntapiwci-a300e9u25</td>
<td>gcsdrsql/db_sc46_copy ANFCVOODemo</td>
<td>17 seconds</td>
<td>idle</td>
<td>snapmirrored</td>
<td>May 6, 2022, 11:43:18 AM 110.59 KB</td>
</tr>
<tr>
<td></td>
<td>gcsdrsql/db-sc46_copy ANFCVOODemo</td>
<td>gcsdrsql/db-sc46_copy ANFCVOODemo</td>
<td>7 seconds</td>
<td>idle</td>
<td>snapmirrored</td>
<td>May 6, 2022, 11:42:20 AM 7.22 MB</td>
</tr>
<tr>
<td></td>
<td>gcsdrsql/log-sc46 ntapiwci-a300e9e25</td>
<td>gcsdrsql/log_sc46_copy ANFCVOODemo</td>
<td>16 seconds</td>
<td>idle</td>
<td>snapmirrored</td>
<td>May 6, 2022, 11:42:52 AM 120.69 KB</td>
</tr>
</tbody>
</table>
Configure AVS hosts and CVO data access

Two important factors to consider when deploying the SDDC are the size of the SDDC cluster in the Azure VMware solution and how long to keep the SDDC in service. These two key considerations for a disaster recovery solution help reduce the overall operational costs. The SDDC can be as small as three hosts, all the way up to a multi-host cluster in a full-scale deployment.

The decision to deploy an AVS cluster is primarily based on the RPO/RTO requirements. With the Azure VMware solution, the SDDC can be provisioned just in time in preparation for either testing or an actual disaster event. An SDDC deployed just in time saves on ESXi host costs when you are not dealing with a disaster. However, this form of deployment affects the RTO by a few of hours while SDDC is being provisioned.

The most common deployed option is to have SDDC running in an always-on, pilot-light mode of operation. This option provides a small footprint of three hosts that are always available, and it also speeds up recovery operations by providing a running baseline for simulation activities and compliance checks, thus avoiding the risk of operational drift between the production and DR sites. The pilot-light cluster can be scaled up quickly to the desired level when needed to handle an actual DR event.

To configure AVS SDDC (be it on-demand or in pilot-light mode), see Deploy and configure the Virtualization Environment on Azure. As a prerequisite, verify that the guest VMs residing on the AVS hosts are able to consume data from Cloud Volumes ONTAP after connectivity has been established.

After Cloud Volumes ONTAP and AVS have been configured properly, begin configuring Jetstream to automate the recovery of on-premises workloads to AVS (VMs with application VMDKs and VMs with in-guest storage) by using the VAIO mechanism and by leveraging SnapMirror for application volumes copies to Cloud Volumes ONTAP.
Install JetStream DR in on-premises datacenter

JetStream DR software consists of three major components: the JetStream DR Management Server Virtual Appliance (MSA), the DR Virtual Appliance (DRVA), and host components (I/O filter packages). The MSA is used to install and configure host components on the compute cluster and then to administer JetStream DR software. The installation process is as follows:

1. Check the prerequisites.
2. Run the Capacity Planning Tool for resource and configuration recommendations.
3. Deploy the JetStream DR MSA to each vSphere host in the designated cluster.
4. Launch the MSA using its DNS name in a browser.
5. Register the vCenter server with the MSA.
6. After JetStream DR MSA has been deployed and the vCenter Server has been registered, navigate to the JetStream DR plug-in with the vSphere Web Client. This can be done by navigating to Datacenter > Configure > JetStream DR.

7. From the JetStream DR interface, complete the following tasks:
   a. Configure the cluster with the I/O filter package.
   b. Add the Azure Blob storage located at the recovery site.
8. Deploy the required number of DR Virtual Appliances (DRVAs) from the Appliances tab.

- Use the capacity planning tool to estimate the number of DRVAs required.

9. Create replication log volumes for each DRVA using the VMDK from the datastores available or the independent shared iSCSI storage pool.
10. From the Protected Domains tab, create the required number of protected domains using information about the Azure Blob Storage site, the DRVA instance, and the replication log. A protected domain defines a specific VM or set of application VMs within the cluster that are protected together and assigned a priority order for failover/failback operations.

11. Select the VMs to be protected and group the VMs into applications groups based on dependency. Application definitions allow you to group sets of VMs into logical groups that contain their boot orders, boot delays, and optional application validations that can be executed upon recovery.

Make sure that the same protection mode is used for all VMs in a protected domain.
Write-Back(VMDK) mode offers higher performance.

12. Make sure that replication log volumes are placed on high-performance storage.

13. After you are done, click Start Protection for the protected domain. This starts data replication for the selected VMs to the designated Blob store.
14. After replication is completed, the VM protection status is marked as Recoverable.

Failover runbooks can be configured to group the VMs (called a recovery group), set the boot order sequence, and modify the CPU/memory settings along with the IP configurations.

15. Click Settings and then click the runbook Configure link to configure the runbook group.

16. Click the Create Group button to begin creating a new runbook group.
If needed, in the lower portion of the screen, apply custom pre-scripts and post-scripts to automatically run prior to and following operation of the runbook group. Make sure that the Runbook scripts are residing on the management server.

17. Edit the VM settings as required. Specify the parameters for recovering the VMs, including the boot sequence, the boot delay (specified in seconds), the number of CPUs, and the amount of memory to allocate. Change the boot sequence of the VMs by clicking the up or down arrows. Options are also provided to Retain MAC.

18. Static IP addresses can be manually configured for the individual VMs of the group. Click the NIC View link of a VM to manually configure its IP address settings.
19. Click the Configure button to save NIC settings for the respective VMs.
The status of both the failover and failback runbooks is now listed as Configured. Failover and failback runbook groups are created in pairs using the same initial group of VMs and settings. If necessary, the settings of any runbook group can be individually customized by clicking its respective Details link and making changes.
Install JetStream DR for AVS in private cloud

A best practice for a recovery site (AVS) is to create a three-node pilot-light cluster in advance. This allows the recovery site infrastructure to be preconfigured, including the following:

- Destination networking segments, firewalls, services like DHCP and DNS, and so on
- Installation of JetStream DR for AVS
- Configuration of ANF volumes as datastores and more

JetStream DR supports a near-zero RTO mode for mission-critical domains. For these domains, destination storage should be preinstalled. ANF is a recommended storage type in this case.

Network configuration including segment creation should be configured on the AVS cluster to match on-premises requirements.

Depending on the SLA and RTO requirements, you can use continuous failover or regular (standard) failover mode. For near-zero RTO, you should start continuous rehydration at the recovery site.

1. To install JetStream DR for AVS on an Azure VMware Solution private cloud, use the Run command. From the Azure portal, go to Azure VMware solution, select the private cloud, and select Run command > Packages > JSDR.Configuration.

   The default CloudAdmin user of the Azure VMware Solution doesn’t have sufficient privileges to install JetStream DR for AVS. The Azure VMware Solution enables simplified and automated installation of JetStream DR by invoking the Azure VMware Solution Run command for JetStream DR.

   The following screenshot shows installation using a DHCP-based IP address.

2. After JetStream DR for AVS installation is complete, refresh the browser. To access the JetStream DR UI, go to SDDC Datacenter > Configure > JetStream DR.
3. From the JetStream DR interface, complete the following tasks:

   a. Add the Azure Blob Storage account that was used to protect the on-premises cluster as a storage site and then run the Scan Domains option.

   b. In the pop-up dialog window that appears, select the protected domain to import and then click its Import link.

4. The domain is imported for recovery. Go to the Protected Domains tab and verify that the intended domain has been selected or choose the desired one from the Select Protected Domain menu. A list of the recoverable VMs in the protected domain is displayed.

5. After the protected domains are imported, deploy DRVA appliances.
These steps can also be automated using CPT-created plans.

6. Create replication log volumes using available vSAN or ANF datastores.

7. Import the protected domains and configure the recovery VA to use an ANF datastore for VM placements.

Make sure that DHCP is enabled on the selected segment and that enough IPs are available. Dynamic IPs are temporarily used while domains are recovering. Each recovering VM (including continuous rehydration) requires an individual dynamic IP. After recovery is complete, the IP is released and can be reused.

8. Select the appropriate failover option (continuous failover or failover). In this example, continuous rehydration (continuous failover) is selected.

Although Continuous Failover and Failover modes differ on when configuration is performed, both failover modes are configured using the same steps. Failover steps are configured and performed together in response to a disaster event. Continuous failover can be configured at any time and then allowed to run in the background during normal system operation. After a disaster event has occurred, continuous failover is completed to immediately transfer ownership of the protected VMs to the recovery site (near-zero RTO).
The continuous failover process begins, and its progress can be monitored from the UI. Clicking the blue icon in the Current Step section exposes a pop-up window showing details of the current step of the failover process.
Failover and Failback

1. After a disaster occurs in the protected cluster of the on-premises environment (partial or complete failure), you can trigger the failover for VMs using Jetstream after breaking the SnapMirror relationship for the respective application volumes.

   ![Failover and Failback Diagram](image)

   - This step can easily be automated to facilitate the recovery process.

2. Access the Jetstream UI on AVS SDDC (destination side) and trigger the failover option to complete failover. The task bar shows progress for failover activities.

   In the dialog window that appears when completing failover, the failover task can be specified as planned or assumed to be forced.
Forced failover assumes the primary site is no longer accessible and ownership of the protected domain should be directly assumed by the recovery site.
3. After continuous failover is complete, a message appears confirming completion of the task. When the task is complete, access the recovered VMs to configure iSCSI or NFS sessions.

The failover mode changes to Running in Failover and the VM status is Recoverable. All the VMs of the protected domain are now running at the recovery site in the state specified by the failover runbook settings.

To verify the failover configuration and infrastructure, JetStream DR can be operated in test mode (Test Failover option) to observe the recovery of virtual machines and their data from the object store into a test recovery environment. When a failover procedure is executed in test mode, its operation resembles an actual failover process.
4. After the virtual machines are recovered, use storage disaster recovery for in-guest storage. To demonstrate this process, SQL server is used in this example.

5. Log into the recovered SnapCenter VM on AVS SDDC and enable DR mode.
   a. Access the SnapCenter UI using the browser.
   b. In the Settings page, navigate to Settings > Global Settings > Disaster Recovery.
   c. Select Enable Disaster Recovery.
   d. Click Apply.
   e. Verify whether the DR job is enabled by clicking Monitor > Jobs.
NetApp SnapCenter 4.6 or later should be used for storage disaster recovery. For previous versions, application-consistent snapshots (replicated using SnapMirror) should be used and manual recovery should be executed in case previous backups must be recovered in the disaster recovery site.

6. Make sure that the SnapMirror relationship is broken.

7. Attach the LUN from Cloud Volumes ONTAP to the recovered SQL guest VM with same drive letters.

8. Open iSCSI Initiator, clear the previous disconnected session and add the new target along with multipath for the replicated Cloud Volumes ONTAP volumes.
9. Make sure that all the disks are connected using the same drive letters that were used prior to DR.

10. Restart the MSSQL server service.
11. Make sure that the SQL resources are back online.

In the case of NFS, attach the volumes using the mount command and update the /etc/fstab entries.

At this point, operations can be run and business continues normally.
On the NSX-T end, a separate dedicated tier-1 gateway can be created for simulating failover scenarios. This ensures that all workloads can communicate with each other but that no traffic can route in or out of the environment, so that any triage, containment, or hardening tasks can be performed without risk of cross-contamination. This operation is outside of the scope of this document, but it can easily be achieved for simulating isolation.

After the primary site is up and running again, you can perform failback. VM protection is resumed by Jetstream and the SnapMirror relationship must be reversed.

1. Restore the on-premises environment. Depending on the type of disaster incident, it might be necessary to restore and/or verify the configuration of the protected cluster. If necessary, JetStream DR software might need to be reinstalled.
2. Access the restored on-premises environment, go to the Jetstream DR UI, and select the appropriate protected domain. After the protected site is ready for failback, select the Failback option in the UI. The CPT-generated failback plan can also be used to initiate the return of the VMs and their data from the object store back to the original VMware environment.

Specify the maximum delay after pausing the VMs in the recovery site and restarting them in the protected site. The time need to complete this process includes the completion of replication after stopping failover VMs, the time needed to clean the recovery site, and the time needed to recreate VMs in the protected site. NetApp recommends 10 minutes.
3. Complete the failback process and then confirm the resumption of VM protection and data consistency.

4. After the VMs are recovered, disconnect the secondary storage from the host and connect to the primary storage.
5. Restart the MSSQL server service.
6. Verify that the SQL resources are back online.

To failback to the primary storage, make sure that the relationship direction remains the same as it was before the failover by performing a reverse resync operation.

To retain the roles of primary and secondary storage after the reverse resync operation, perform the reverse resync operation again.

This process is applicable to other applications like Oracle, similar database flavors, and any other applications using guest-connected storage.
As always, test the steps involved for recovering the critical workloads before porting them into production.

**Benefits of this solution**

- Uses the efficient and resilient replication of SnapMirror.
- Recovers to any available points in time with ONTAP snapshot retention.
- Full automation is available for all required steps to recover hundreds to thousands of VMs, from the storage, compute, network, and application validation steps.
- SnapCenter uses cloning mechanisms that do not change the replicated volume.
  - This avoids the risk of data corruption for volumes and snapshots.
  - Avoids replication interruptions during DR test workflows.
  - Leverages the DR data for workflows beyond DR, such as dev/test, security testing, patch and upgrade testing, and remediation testing.
- CPU and RAM optimization can help lower cloud costs by enabling recovery to smaller compute clusters.

**TR-4955: Disaster Recovery with Azure NetApp Files (ANF) and Azure VMware Solution (AVS)**

**Author(s): Niyaz Mohamed, NetApp Solutions Engineering**

**Overview**

Disaster recovery using block-level replication between regions within the cloud is a resilient and cost-effective way of protecting the workloads against site outages and data corruption events (for example, ransomware). With Azure NetApp files (ANF) cross-region volume replication, VMware workloads running on an Azure VMware Solution (AVS) SDDC site using Azure NetApp files volumes as an NFS datastore on the primary AVS site can be replicated to a designated secondary AVS site in the target recovery region.

Disaster Recovery Orchestrator (DRO) (a scripted solution with a UI) can be used to seamlessly recover workloads replicated from one AVS SDDC to another. DRO automates recovery by breaking replication peering and then mounting the destination volume as a datastore, through VM registration to AVS, to network mappings directly on NSX-T (included with all AVS private clouds).
Prerequisites and general recommendations

- Verify that you have enabled cross-region replication by creating replication peering. See [Create volume replication for Azure NetApp Files](#).
- You must configure ExpressRoute Global Reach between the source and target Azure VMware Solution private clouds.
- You must have a service principal that can access resources.
- The following topology is supported: primary AVS site to secondary AVS site.
- Configure the replication schedule for each volume appropriately based on business needs and the data-change rate.

Cascading and fan-in and fan-out topologies are not supported.

Getting started

**Deploy Azure VMware Solution**

The Azure VMware Solution (AVS) is a hybrid cloud service that provides fully functional VMware SDDCs within a Microsoft Azure public cloud. AVS is a first-party solution fully managed and supported by Microsoft and verified by VMware that uses Azure infrastructure. Therefore, customers get VMware ESXi for compute virtualization, vSAN for hyper-converged storage, and NSX for networking and security, all while taking advantage of Microsoft Azure’s global presence, class-leading data-center facilities, and proximity to the rich ecosystem of native Azure services and solutions. A combination of Azure VMware Solution SDDC and Azure NetApp Files provides the best performance with minimal network latency.

To configure an AVS private cloud on Azure, follow the steps in this link for NetApp documentation and in this link for Microsoft documentation. A pilot-light environment set up with a minimal configuration can be used for DR purposes. This setup only contains core components to support critical applications, and it can scale out and spawn more hosts to take the bulk of the load if a failover occurs.
In the initial release, DRO supports an existing AVS SDDC cluster. On-demand SDDC creation will be available in an upcoming release.

Provision and configure Azure NetApp Files

Azure NetApp Files is a high-performance, enterprise-class, metered file-storage service. Follow the steps in this link to provision and configure Azure NetApp Files as a NFS datastore to optimize AVS private cloud deployments.

Create volume replication for Azure NetApp Files-powered datastore volumes

The first step is to set up cross-region replication for the desired datastore volumes from the AVS primary site to the AVS secondary site with the appropriate frequencies and retentions.

Follow the steps in this link to set up cross-region replication by creating replication peering. The service level for the destination capacity pool can match that of the source capacity pool. However, for this specific use case, you can select the standard service level and then modify the service level in the event of a real disaster or DR simulations.

A cross-region replication relationship is a prerequisite and must be created beforehand.

DRO installation

To get started with DRO, use the Ubuntu operating system on the designated Azure virtual machine and make sure you meet the prerequisites. Then install the package.

Prerequisites:

- Service principal that can access resources.
- Make sure that appropriate connectivity exists to the source and destination SDDC and Azure NetApp Files instances.
- DNS resolution should be in place if you are using DNS names. Otherwise, use IP addresses for vCenter.

OS requirements:

- Ubuntu Focal 20.04 (LTS) The following packages must be installed on the designated agent virtual machine:
  - Docker
  - Docker-compose
  - JqChange

Modify docker.sock to this new permission: `sudo chmod 666 /var/run/docker.sock`.

The `deploy.sh` script executes all required prerequisites.
The steps are as follows:

1. Download the installation package on the designated virtual machine:

   ```
git clone https://github.com/NetApp/DRO-Azure.git
   ```

   The agent must be installed in the secondary AVS site region or in the primary AVS site region in a separate AZ than the SDDC.

2. Unzip the package, run the deployment script, and enter the host IP (for example, 10.10.10.10).

   ```
tar xvf draas_package.tar
Navigating to the directory and running the deploy script as below:
sudo sh deploy.sh
   ```

3. Access the UI using the following credentials:

   - **Username:** admin
   - **Password:** admin

![NetApp DRO UI](image)

**DRO configuration**

After Azure NetApp Files and AVS have been configured properly, you can begin configuring DRO to automate the recovery of workloads from the primary AVS site to the secondary AVS site. NetApp recommends deploying the DRO agent in the secondary AVS site and configuring the ExpressRoute gateway connection so that the DRO agent can communicate via the network with the appropriate AVS and Azure NetApp Files components.

The first step is to Add credentials. DRO requires permission to discover Azure NetApp Files and the Azure VMware Solution. You can grant the required permissions to an Azure account by creating and setting up an
Azure Active Directory (AD) application and by obtaining the Azure credentials that DRO needs. You must bind the service principal to your Azure subscription and assign it a custom role that has the relevant required permissions. When you add source and destination environments, you are prompted to select the credentials associated with the service principal. You need to add these credentials to DRO before you can click Add New Site.

To perform this operation, complete the following steps:

1. Open DRO in a supported browser and use the default username and password (admin/admin). The password can be reset after the first login using the Change Password option.

2. In the upper right of the DRO console, click the **Settings** icon, and select **Credentials**.

3. Click Add New Credential and follow the steps in the wizard.

4. To define the credentials, enter information about the Azure Active Directory service principal that grants the required permissions:
   - Credential name
   - Tenant ID
   - Client ID
   - Client secret
   - Subscription ID

   You should have captured this information when you created the AD application.

5. Confirm the details about the new credentials and click Add Credential.

After you add the credentials, it’s time to discover and add the primary and secondary AVS sites (both vCenter and the Azure NetApp files storage account) to DRO. To add the source and destination site, complete the following steps:

6. Go to the **Discover** tab.

7. Click **Add New Site**.
8. Add the following primary AVS site (designated as **Source** in the console).
   - SDDC vCenter
   - Azure NetApp Files storage account

9. Add the following secondary AVS site (designated as **Destination** in the console).
   - SDDC vCenter
   - Azure NetApp Files storage account

10. Add site details by clicking **Source**, entering a friendly site name, and select the connector. Then click **Continue**.

    For demonstration purposes, adding a source site is covered in this document.

11. Update the vCenter details. To do this, select the credentials, Azure region, and resource group from the dropdown for the primary AVS SDDC.

12. DRO lists all the available SDDCs within the region. Select the designated private cloud URL from the dropdown.

13. Enter the `cloudadmin@vsphere.local` user credentials. This can be accessed from Azure Portal. Follow the steps mentioned in this link. Once done, click **Continue**.
14. Select the Source Storage details (ANF) by selecting the Azure Resource group and NetApp account.

15. Click **Create Site**.

Once added, DRO performs automatic discovery and displays the VMs that have corresponding cross-region replicas from the source site to the destination site. DRO automatically detects the networks and segments used by the VMs and populates them.
The next step is to group the required VMs into their functional groups as resource groups.

**Resource groupings**

After the platforms have been added, group the VMs you want to recover into resource groups. DRO resource groups allow you to group a set of dependent VMs into logical groups that contain their boot orders, boot delays, and optional application validations that can be executed upon recovery.

To start creating resource groups, click the **Create New Resource Group** menu item.

1. Access **Resource Groups** and click **Create New Resource Group**.

2. Under New Resource Group, select the source site from the dropdown and click **Create**.

3. Provide the resource group details and click **Continue**.

4. Select appropriate VMs using the search option.

5. Select the **Boot Order** and **Boot Delay** (secs) for all the selected VMs. Set the order of the power-on sequence by selecting each virtual machine and setting up the priority for it. The default value for all virtual machines is 3. The options are as follows:
   - The first virtual machine to power on
   - Default
The last virtual machine to power on

6. Click **Create Resource Group**.

Replication plans

You must have a plan to recover applications in the event of a disaster. Select the source and destination vCenter platforms from the drop down, pick the resource groups to be included in this plan, and also include the grouping of how applications should be restored and powered on (for example, domain controllers, tier-1, tier-2, and so on). Plans are often called blueprints as well. To define the recovery plan, navigate to the Replication Plan tab, and click **New Replication Plan**.

To start creating a replication plan, complete the following steps:

1. Navigate to **Replication Plans** and click **Create New Replication Plan**.

2. On the **New Replication Plan**, provide a name for the plan and add recovery mappings by selecting the Source Site, associated vCenter, Destination Site, and associated vCenter.
3. After recovery mapping is complete, select the **Cluster Mapping**.

4. Select **Resource Group Details** and click **Continue**.

5. Set the execution order for the resource group. This option enables you to select the sequence of operations when multiple resource groups exist.

6. Once done, set network mapping to the appropriate segment. The segments should already be provisioned on the secondary AVS cluster, and, to map the VMs to those, select the appropriate segment.

7. Datastore mappings are automatically selected based on the selection of VMs.

Cross-region replication (CRR) is at the volume level. Therefore, all VMs residing on the respective volume are replicated to the CRR destination. Make sure to select all VMs that are part of the datastore, because only virtual machines that are part of the replication plan are processed.
8. Under VM details, you can optionally resize the VMs CPU and RAM parameters. This can be very helpful when you are recovering large environments to smaller target clusters or when you are conducting DR tests without having to provision a one-to-one physical VMware infrastructure. Also, modify the boot order and boot delay (secs) for all the selected VMs across the resource groups. There is an additional option to modify the boot order if any changes are required from what you selected during resource-group boot-order selection. By default, the boot order selected during resource-group selection is used, however any modifications can be performed at this stage.

9. Click **Create Replication Plan**. After the replication plan is created, you can exercise the failover, test failover, or migrate options depending on your requirements.
During the failover and test failover options, the most recent snapshot is used, or a specific snapshot can be selected from a point-in-time snapshot. The point-in-time option can be very beneficial if you are facing a corruption event like ransomware, where the most recent replicas are already compromised or encrypted. DRO shows all available time points.

To trigger failover or test failover with the configuration specified in the replication plan, you can click Failover or Test Failover. You can monitor the replication plan in the task menu.
After failover is triggered, the recovered items can be seen in the secondary site AVS SDDC vCenter (VMs, networks, and datastores). By default, the VMs are recovered to Workload folder.

Failback can be triggered at the replication plan level. In case of test failover, the tear down option can be used to roll back the changes and remove the newly created volume. Failbacks related to failover are a two-step process. Select the replication plan and select **Reverse Data sync**.
After this step is complete, trigger failback to move back to the primary AVS site.

From the Azure portal, we can see that the replication health has been broken off for the appropriate volumes that were mapped to the secondary site AVS SDDC as read/write volumes. During test failover, DRO does not map the destination or replica volume. Instead, it creates a new volume of the required cross-region replication snapshot and exposes the volume as a datastore, which consumes additional physical capacity from the capacity pool and ensures that the source volume is not modified. Notably, replication jobs can continue during DR tests or triage workflows. Additionally, this process makes sure that the recovery can be cleaned up without the risk of the replica being destroyed if errors occur or corrupted data is recovered.

**Ransomware recovery**

Recovering from ransomware can be a daunting task. Specifically, it can be difficult for IT organizations to pinpoint what the safe point of return is, and, once that’s determined, how to ensure that recovered workloads are safeguarded from the attacks reoccurring (for example, from sleeping malware or through vulnerable applications).

DRO addresses these concerns by allowing organizations to recover from any available point-in-time. Workloads are then recovered to functional and yet isolated networks, so that applications can function and communicate with each other but are not exposed to any north-south traffic. This process gives security teams...
a safe place to conduct forensics and identify any hidden or sleeping malware.

Conclusion

The Azure NetApp Files and Azure VMware disaster recovery solution provide you with the following benefits:

- Leverage efficient and resilient Azure NetApp Files cross-region replication.
- Recover to any available point-in-time with snapshot retention.
- Fully automate all required steps to recover hundreds to thousands of VMs from the storage, compute, network, and application validation steps.
- Workload recovery leverages the “Create new volumes from the most recent snapshots” process, which doesn’t manipulate the replicated volume.
- Avoid any risk of data corruption on the volumes or snapshots.
- Avoid replication interruptions during DR test workflows.
- Leverage DR data and cloud compute resources for workflows beyond DR, such as dev/test, security testing, patch and upgrade testing, and remediation testing.
- CPU and RAM optimization can help lower cloud costs by allowing recovery to smaller compute clusters.

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

- Create volume replication for Azure NetApp Files

- Cross-region replication of Azure NetApp Files volumes

- Azure VMware Solution

- Deploy and configure the Virtualization Environment on Azure

- Deploy and configure Azure VMware Solution

Using Veeam Replication and Azure NetApp Files datastore for disaster recovery to Azure VMware Solution

Author: Niyaz Mohamed - NetApp Solutions Engineering
Overview

Azure NetApp Files (ANF) datastores decouples storage from compute and unlocks the flexibility needed for any organisation to take their workloads to the cloud. It provides customers with flexible, high-performance storage infrastructure that scales independently of compute resources. Azure NetApp Files datastore’s simplifies and optimizes the deployment alongside Azure VMware Solution (AVS) as a disaster recovery site for on premises VMWare environments.

Azure NetApp Files (ANF) volume based NFS datastores can be used to replicate data from on-premises using any validated third-party solution that provides VM replication capability. By adding Azure NetApp Files datastores, it will enable cost optimised deployment vs building an Azure VMware Solution SDDC with enormous amount of ESXi hosts to accommodate the storage. This approach is called a “Pilot Light Cluster”. A pilot light cluster is a minimal AVS host configuration (3 x AVS nodes) along with Azure NetApp Files Datastore capacity.

The objective is to maintain a low-cost infrastructure with all the core components to handle a failover. A pilot light cluster can scale out and provision more AVS hosts if a failover does occur. And once the failover is complete and normal operations are restored, the pilot light cluster can scale back down to low-cost mode of operations.

Purposes of this document

This article describes how to use Azure NetApp Files datastore with Veeam Backup and replication to set up disaster recovery for on-premises VMware VMs to (AVS) using the Veeam VM replication software functionality.

Veeam Backup & Replication is a backup and replication application for virtual environments. When virtual machines are replicated, Veeam Backup & Replication is replicated from on AVS, the software will create an exact copy of the VMs in the native VMware vSphere format on the target AVS SDDC cluster. Veeam Backup & Replication will keep the copy synchronized with the original VM. Replication provides the best recovery time objective (RTO) as there is a mounted copy of a VM at the DR site in a ready-to-start state.

This replication mechanism ensures that the workloads can quickly start in a AVS SDDC in the case of a disaster event. The Veeam Backup & Replication software also optimizes traffic transmission for replication over WAN and slow connections. In addition, it also filters out duplicate data blocks, zero data blocks, swap files, and “excluded VM guest OS files”. The software will also compress the replica traffic. To prevent replication jobs from consuming the entire network bandwidth, WAN accelerators and network throttling rules can be utilized.

The replication process in Veeam Backup & Replication is job driven which means replication is performed by configuring replication jobs. In the case of a disaster event, failover can be triggered to recover the VMs by failing over to its replica copy. When failover is performed, a replicated VM takes over the role of the original VM. Failover can be performed to the latest state of a replica or to any of its good known restore points. This enables ransomware recovery or isolated testing as needed. Veeam Backup & Replication offers multiple options to handle different disaster recovery scenarios.

Solution Deployment

High level steps

1. Veeam Backup and Replication software is running in an on-premises environment with appropriate network connectivity.

2. Deploy Azure VMware Solution (AVS) private cloud and attach Azure NetApp Files datastores to Azure
VMware Solution hosts.

A pilot-light environment set up with a minimal configuration can be used for DR purposes. VMs will fail over to this cluster in the event of an incident, and additional nodes can be added).

3. Set up replication job to create VM replicas using Veeam Backup and Replication.
4. Create failover plan and perform failover.
5. Switch back to production VMs once the disaster event is complete and primary site is Up.

**Pre-requisites for Veeam VM Replication to AVS and ANF datastores**

1. Ensure the Veeam Backup & Replication backup VM is connected to the source as well as the target AVS SDDC clusters.
2. The backup server must be able to resolve short names and connect to source and target vCenters.
3. The target Azure NetApp Files datastore must have enough free space to store VMDKs of replicated VMs.

For additional information, refer to "Considerations and Limitations" covered [here](#).

**Deployment Details**
Step 1: Replicate VMs

Veeam Backup & Replication leverages VMware vSphere snapshot capabilities. During replication, Veeam Backup & Replication requests VMware vSphere to create a VM snapshot. The VM snapshot is the point-in-time copy of a VM that includes virtual disks, system state, configuration and metadata. Veeam Backup & Replication uses the snapshot as a source of data for replication.

To replicate VMs, follow the below steps:

1. Open the Veeam Backup & Replication Console.
2. On the Home view, right click the jobs node and select Replication Job > Virtual machine.
3. Specify a job name and select the appropriate advanced control checkbox. Click Next.
   - Select the Replica seeding check box if connectivity between on-premises and Azure has restricted bandwidth.
   - Select the Network remapping (for AVS SDDC sites with different networks) check box if segments on Azure VMware Solution SDDC do not match that of on-premises site networks.
   - If the IP addressing scheme in on-premises production site differs from the scheme in the target AVS site, select the Replica re-IP (for DR sites with different IP addressing scheme) check box.

4. Select the VMs to be replicated to Azure NetApp Files datastore attached to an Azure VMware Solution SDDC in the Virtual Machines step. The Virtual machines can be placed on vSAN to fill the available vSAN datastore capacity. In a pilot light cluster, the usable capacity of a 3-node cluster will be limited. The rest of the data can be easily placed on Azure NetApp Files datastores so that the VMs can recovered, and cluster can be expanded to meet the CPU/mem requirements. Click Add, then in the Add Object window select the necessary VMs or VM containers and click Add. Click Next.

5. After that, select the destination as Azure VMware Solution SDDC cluster / host and the appropriate resource pool, VM folder and FSx for ONTAP datastore for VM replicas. Then click Next.

6. In the next step, create the mapping between source and destination virtual network as needed.

7. In the Job Settings step, specify the backup repository that will store metadata for VM replicas, retention policy and so on.

8. Update the Source and Target proxy servers in the Data Transfer step and leave Automatic selection (default) and keep Direct option selected and click Next.

9. At the Guest Processing step, select Enable application-aware processing option as needed. Click Next.

10. Choose the replication schedule to run the replication job to run on a regular basis.
11. At the **Summary** step of the wizard, review details of the replication job. To start the job right after the wizard is closed, select the **Run the job when I click Finish** check box, otherwise leave the check box unselected. Then click **Finish** to close the wizard.

Once the replication job starts, the VMs with the suffix specified will be populated on the destination AVS SDDC cluster / host.

For additional information for Veeam replication, refer [How Replication Works](#).

### Step 2: Create a failover plan

When the initial replication or seeding is complete, create the failover plan. Failover plan helps in performing failover for dependent VMs one by one or as a group automatically. Failover plan is the blueprint for the order in which the VMs are processed including the boot delays. The failover plan also helps to ensure that critical dependant VMs are already running.

To create the plan, navigate to the new sub section called **Replicas** and select **Failover Plan**. Choose the appropriate VMs. Veeam Backup & Replication will look for the closest restore points to this point in time and use them to start VM replicas.

- The failover plan can only be added once the initial replication is complete and the VM replicas are in Ready state.
- The maximum number of VMs that can be started simultaneously when running a failover plan is 10
- During the failover process, the source VMs will not be powered off

To create the **Failover Plan**, do the following:


2. Next provide a name and a description to the plan. Pre and Post-failover script can be added as required. For instance, run a script to shutdown VMs before starting the replicated VMs.

3. Add the VMs to the plan and modify the VM boot order and boot delays to meet the application dependencies.

For additional information for creating replication jobs, refer [Creating Replication Jobs](#).
Step 3: Run the failover plan

During failover, the source VM in the production site is switched over to its replica at the disaster recovery site. As part of the failover process, Veeam Backup & Replication restores the VM replica to the required restore point and moves all I/O activities from the source VM to its replica. Replicas can be used not only in case of a disaster, but also to simulate DR drills. During failover simulation, the source VM remains running. Once all the necessary tests have been conducted, you can undo the failover and return to normal operations.

- Make sure network segmentation is in place to avoid IP conflicts during failover.

To start the failover plan, simply click in Failover Plans tab and right click on your failover plan. Select *Start*. This will failover using the latest restore points of VM replicas. To fail over to specific restore points of VM replicas, select Start to.

The state of the VM replica changes from Ready to Failover and VMs will start on the destination Azure VMware Solution (AVS) SDDC cluster / host.

Once the failover is complete, the status of the VMs will change to “Failover”.

Veeam Backup & Replication stops all replication activities for the source VM until its replica is returned to the Ready state.

For detailed information about failover plans, refer Failover Plans.
Step 4: Failback to the Production site

When the failover plan is running, it is considered as an intermediate step and needs to be finalized based on the requirement. The options include the following:

- **Failback to production** - switch back to the original VM and transfer all changes that took place while the VM replica was running to the original VM.

  When you perform failback, changes are only transferred but not published. Choose **Commit failback** (once the original VM is confirmed to work as expected) or **Undo failback** to get back to the VM replica if the original VM is not working as expected.

- **Undo failover** - switch back to the original VM and discard all changes made to the VM replica while it was running.

- **Permanent Failover** - permanently switch from the original VM to a VM replica and use this replica as the original VM.

In this demo, Failback to production was chosen. Failback to the original VM was selected during the Destination step of the wizard and “Power on VM after restoring” check box was enabled.

Failback commit is one of the ways to finalize failback operation. When failback is committed, it confirms that the changes sent to the VM which is failed back (the production VM) are working as expected. After the commit operation, Veeam Backup & Replication resumes replication activities for the production VM.

For detailed information about the failback process, refer Veeam documentation for **Failover and Failback for replication**.

After failback to production is successful, the VMs are all restored back to the original production site.

**Conclusion**

Azure NetApp Files datastore capability enables Veeam or any validated third-party tool to provide a low-cost DR solution by leveraging Pilot light clusters instead of standing up a large cluster only to accommodate VM replicas. This provides an efficacious way to handle a tailored, customized disaster recovery plan and to reuse existing backup products in house for DR, enabling cloud-based disaster recovery by exiting on-premises DR datacenters. It is possible to failover by clicking a button in case of disaster or to failover automatically if a disaster occurs.

To learn more about this process, feel free to follow the detailed walkthrough video.

https://netapp.hosted.panopto.com/Panopto/Pages/Embed.aspx?id=2855e0d5-97e7-430f-944a-b061015e9278
Migrating Workloads on Azure / AVS

TR-4940: Migrate workloads to Azure NetApp Files datastore using VMware HCX - Quickstart guide

Author(s): NetApp Solutions Engineering

Overview: Migrating virtual machines with VMware HCX, Azure NetApp Files datastores, and Azure VMware solution

One of the most common use cases for the Azure VMware Solution and Azure NetApp Files datastore is the migration of VMware workloads. VMware HCX is a preferred option and provides various migration mechanisms to move on-premises virtual machines (VMs) and its data to Azure NetApp Files datastores.

VMware HCX is primarily a migration platform that is designed to simplify application migration, workload rebalancing, and even business continuity across clouds. It is included as part of Azure VMware Solution Private Cloud and offers many ways to migrate workloads and can be used for disaster recovery (DR) operations.

This document provides step-by-step guidance for provisioning Azure NetApp Files datastore followed by downloading, deploying, and configuring VMware HCX, including all its main components in on-premises and the Azure VMware Solution side including Interconnect, Network Extension, and WAN optimization for enabling various VM migration mechanisms.

VMware HCX works with any datastore type as the migration is at the VM level. Hence this document is applicable to existing NetApp customers and non-NetApp customers who are planning to deploy Azure NetApp Files with Azure VMware Solution for a cost-effective VMware cloud deployment.

High-level steps

This list provides the high-level steps necessary to install and configure HCX Cloud Manager on the Azure cloud side and install HCX Connector on-premises:

1. Install HCX through the Azure portal.
2. Download and deploy the HCX Connector Open Virtualization Appliance (OVA) installer in the on-premises VMware vCenter Server.
3. Activate HCX with the license key.
4. Pair the on-premises VMware HCX Connector with Azure VMware Solution HCX Cloud Manager.
5. Configure the network profile, compute profile, and service mesh.
6. (Optional) Perform network extension to avoid re-IP during migrations.
7. Validate the appliance status and ensure that migration is possible.
8. Migrate the VM workloads.
Before you begin, make sure the following prerequisites are met. For more information, see this link. After the prerequisites, including connectivity, are in place, configure and activate HCX by generating the license key from the Azure VMware Solution portal. After the OVA installer is downloaded, proceed with the installation process as described below.

HCX advanced is the default option and VMware HCX Enterprise edition is also available through a support ticket and supported at no additional cost.

- Use an existing Azure VMware solution software-defined data center (SDDC) or create a private cloud by using this NetApp link or this Microsoft link.
- Migration of VMs and associated data from the on-premises VMware vSphere-enabled data center requires network connectivity from the data center to the SDDC environment. Before migrating workloads, set up a site-to-site VPN or Express route global reach connection between the on-premises environment and the respective private cloud.
- The network path from on-premises VMware vCenter Server environment to the Azure VMware Solution private cloud must support the migration of VMs by using vMotion.
- Make sure the required firewall rules and ports are allowed for vMotion traffic between the on-premises vCenter Server and SDDC vCenter. On the private cloud, routing on the vMotion network is configured by default.
- Azure NetApp Files NFS volume should be mounted as a datastore in Azure VMware Solution. Follow the steps detailed in this link to attach Azure NetApp Files datastores to Azure VMware Solutions hosts.
High Level Architecture

For testing purposes, the lab environment from on-premises used for this validation was connected through a site-to-site VPN, which allows on-premises connectivity to Azure VMware Solution.

Solution Deployment

Follow the series of steps to complete the deployment of this solution:
**Step 1: Install HCX through Azure Portal using the Add-ons option**

To perform the installation, complete the following steps:

1. Log in to the Azure Portal and access the Azure VMware Solution private cloud.
2. Select the appropriate private cloud and access Add-ons. This can be done by navigating to **Manage > Add-ons**.
3. In the HCX Workload Mobility section, click **Get Started**.

   ![HCX installation in Azure Portal](image)

1. Select the **I Agree with Terms and Conditions** option and click **Enable and Deploy**.

   - The default deployment is HCX Advanced. Open a support request to enable the Enterprise edition.

   - The deployment takes approximately 25 to 30 minutes.
Step 2: Deploy the installer OVA in the on-premises vCenter Server

For the on-premises Connector to connect to the HCX Manager in Azure VMware Solution, make sure the appropriate firewall ports are open in the on-premises environment.

To download and install HCX Connector in the on-premises vCenter Server, complete the following steps:

1. From the Azure portal, go to the Azure VMware Solution, select the private cloud, and select Manage > Add-ons > Migration using HCX and copy the HCX Cloud Manager portal to download the OVA file.

   Use the default CloudAdmin user credentials to access the HCX portal.

   ![CloudAdmin login](image)

   1. After you access the HCX portal with `cloudadmin@vsphere.local` using the jump host, navigate to Administration > System Updates and click Request Download Link.

   Either download or copy the link to the OVA and paste it into a browser to begin the download process of the VMware HCX Connector OVA file to deploy on the on-premises vCenter Server.
1. After the OVA is downloaded, deploy it on to the on-premises VMware vSphere environment by using the **Deploy OVF Template** option.

1. Enter all the required information for the OVA deployment, click **Next**, and then click **Finish** to deploy the VMware HCX connector OVA.

   - **Power on the virtual appliance manually.**

For step-by-step instructions, see the **VMware HCX User Guide**.
Step 3: Activate HCX Connector with the license key

After you deploy the VMware HCX Connector OVA on-premises and start the appliance, complete the following steps to activate HCX Connector. Generate the license key from the Azure VMware Solution portal and activate it in VMware HCX Manager.

1. From the Azure portal, go to the Azure VMware Solution, select the private cloud, and select Manage > Add-ons > Migration using HCX.
2. Under Connect with on-premise Using HCX keys, click Add and copy the activation key.

A separate key is required for each on-premises HCX Connector that is deployed.

1. Log into the on-premises VMware HCX Manager at https://hcxmanagerIP:9443 using administrator credentials.
   - Use the password defined during the OVA deployment.

1. In the licensing, enter the key copied from step 3 and click Activate.
   - The on-premises HCX Connector should have internet access.

1. Under Datacenter Location, provide the nearest location for installing the VMware HCX Manager on-premises. Click Continue.

1. Under System Name, update the name and click Continue.

1. Click Yes, Continue.

1. Under Connect your vCenter, provide the fully qualified domain name (FQDN) or IP address of vCenter Server and the appropriate credentials and click Continue.
   - Use the FQDN to avoid connectivity issues later.
1. Under **Configure SSO/PSC**, provide the Platform Services Controller's FQDN or IP address and click **Continue**.

   Enter the VMware vCenter Server FQDN or IP address.

1. Verify that the information entered is correct and click **Restart**.

2. After the services restart, vCenter Server is displayed as green on the page that appears. Both vCenter Server and SSO must have the appropriate configuration parameters, which should be the same as the previous page.

   This process should take approximately 10 to 20 minutes and for the plug-in to be added to the vCenter Server.
Step 4: Pair on-premises VMware HCX Connector with Azure VMware Solution HCX Cloud Manager

After HCX Connector is installed in both on-premises and Azure VMware Solution, configure the on-premises VMware HCX Connector for Azure VMware Solution private cloud by adding the pairing. To configure the site pairing, complete the following steps:

1. To create a site pair between the on-premises vCenter environment and Azure VMware Solution SDDC, log in to the on-premises vCenter Server and access the new HCX vSphere Web Client plug-in.

1. Under Infrastructure, click **Add a Site Pairing**.

   Enter the Azure VMware Solution HCX Cloud Manager URL or IP address and the credentials for CloudAdmin role for accessing the private cloud.

1. Click **Connect**.

   VMware HCX Connector must be able to route to HCX Cloud Manager IP over port 443.
1. After the pairing is created, the newly configured site pairing is available on the HCX Dashboard.
Step 5: Configure the network profile, compute profile, and service mesh

The VMware HCX Interconnect service appliance provides replication and vMotion-based migration capabilities over the internet and private connections to the target site. The interconnect provides encryption, traffic engineering, and VM mobility. To create an Interconnect service appliance, complete the followings steps:

1. Under Infrastructure, select **Interconnect > Multi-Site Service Mesh > Compute Profiles > Create Compute Profile**.

The compute profiles define the deployment parameters including the appliances that are deployed and which portion of the VMware data center are accessible to HCX service.

1. After the compute profile is created, create the network profiles by selecting **Multi-Site Service Mesh > Network Profiles > Create Network Profile**.

The network profile defines a range of IP address and networks that are used by HCX for its virtual appliances.

This step requires two or more IP addresses. These IP addresses are assigned from the management network to the Interconnect Appliances.
1. At this time, the compute and network profiles have been successfully created.

2. Create the Service Mesh by selecting the **Service Mesh** tab within the **Interconnect** option and select the on-premises and Azure SDDC sites.

3. The Service Mesh specifies a local and remote compute and network profile pair.

   As part of this process, the HCX appliances are deployed and automatically configured on both the source and target sites in order to create a secure transport fabric.

1. This is the final step of configuration. This should take close to 30 minutes to complete the deployment. After the service mesh is configured, the environment is ready with the IPsec tunnels successfully created to migrate the workload VMs.
Step 6: Migrate workloads

Workloads can be migrated bidirectionally between on-premises and Azure SDDCs using various VMware HCX migration technologies. VMs can be moved to and from VMware HCX-activated entities using multiple migration technologies such as HCX bulk migration, HCX vMotion, HCX Cold migration, HCX Replication Assisted vMotion (available with HCX Enterprise edition), and HCX OS Assisted Migration (available with the HCX Enterprise edition).

To learn more about various HCX migration mechanisms, see VMware HCX Migration Types.

Bulk migration

This section details the bulk migration mechanism. During a bulk migration, the bulk migration capability of HCX uses vSphere Replication to migrate disk files while recreating the VM on the destination vSphere HCX instance.

To initiate bulk VM migrations, complete the following steps:

1. Access the Migrate tab under Services > Migration.

    1. Under Remote Site Connection, select the remote site connection and select the source and destination. In this example, the destination is Azure VMware Solution SDDC HCX endpoint.

    2. Click Select VMs for Migration. This provides a list of all the on-premises VMs. Select the VMs based on the match:value expression and click Add.

    3. In the Transfer and Placement section, update the mandatory fields (Cluster, Storage, Destination, and Network), including the migration profile, and click Validate.
1. After the validation checks are complete, click **Go** to initiate the migration.

During this migration, a placeholder disk is created on the specified Azure NetApp Files datastore within the target vCenter to enable replication of the source VM disk’s data to the placeholder disks. HBR is triggered for a full sync to the target, and after the baseline is complete, an incremental sync is performed based on the recovery point objective (RPO) cycle. After the full/incremental sync is complete, switchover is triggered automatically unless a specific schedule is set.

1. After the migration is complete, validate the same by accessing the destination SDDC vCenter.
For additional and detailed information about various migration options and on how to migrate workloads from on-premises to Azure VMware Solution using HCX, see VMware HCX User Guide.

To learn more about this process, feel free to watch the following video:

Workload Migration using HCX

Here is a screenshot of HCX vMotion option.

To learn more about this process, feel free to watch the following video:

HCX vMotion

- Make sure sufficient bandwidth is available to handle the migration.
- The target ANF datastore should have sufficient space to handle the migration.

Conclusion

Whether you’re targeting all-cloud or hybrid cloud and data residing on any type/vendor storage in on-premises, Azure NetApp Files and HCX provide excellent options to deploy and migrate the application workloads while reducing the TCO by making the data requirements seamless to the application layer.
Whatever the use case, choose Azure VMware Solution along with Azure NetApp Files for rapid realization of cloud benefits, consistent infrastructure, and operations across on-premises and multiple clouds, bidirectional portability of workloads, and enterprise-grade capacity and performance. It is the same familiar process and procedures used to connect the storage and migrate VMs using VMware vSphere Replication, VMware vMotion, or even network file copy (NFC).

**Takeaways**

The key points of this document include:

- You can now use Azure NetApp Files as a datastore on Azure VMware Solution SDDC.
- You can easily migrate data from on-premises to Azure NetApp Files datastore.
- You can easily grow and shrink the Azure NetApp Files datastore to meet the capacity and performance requirements during migration activity.

**Where to find additional information**

To learn more about the information described in this document, refer to the following website links:

- Azure VMware Solution documentation
- Azure NetApp Files documentation
- VMware HCX User Guide

**Region Availability – Supplemental NFS datastore for ANF**

Unresolved directive in ehc/azure-regions.adoc - include::../include/azure-region-support.adoc[]

**NetApp Hybrid Multicloud with VMware Solutions**

**Protecting Workloads on GCP / GCVE**

**Application Consistent Disaster Recovery with NetApp SnapCenter and Veeam Replication**

Authors: Suresh Thoppay, NetApp

**Overview**

Many customers are looking for an effective disaster recovery solution for their application VMs hosted on VMware vSphere. Many of them use their existing backup solution to perform recovery during disaster. Many times that solution increase the RTO and doesn’t meet their expectations. To reduce the RPO and RTO, Veeam VM replication can be utilized even from on-prem to GCVE as long as network connectivity and environemnt with appropriate permissions are available.
NOTE: Veeam VM Replication doesn’t protect VM guest connected storage devices like iSCSI or NFS mounts inside the guest VM. Need to protect those separately.

For application consistent replication for SQL VM and to reduce the RTO, we used SnapCenter to orchestrate snapmirror operations of SQL database and log volumes.

This document provides a step-by-step approach for setting up and performing disaster recovery that uses NetApp SnapMirror, Veeam, and the Google Cloud VMware Engine (GCVE).

**Assumptions**

This document focuses on in-guest storage for application data (also known as guest connected), and we assume that the on-premises environment is using SnapCenter for application-consistent backups.

This document applies to any third-party backup or recovery solution. Depending on the solution used in the environment, follow best practices to create backup policies that meet organizational SLAs.

For connectivity between the on-premises environment and the Google Cloud network, use the connectivity options like dedicated interconnect or Cloud VPN. Segments should be created based on the on-premises VLAN design.

There are multiple options for connecting on-premises datacenters to Google Cloud, which prevents us from outlining a specific workflow in this document. Refer to the Google Cloud documentation for the appropriate on-premises-to-Google connectivity method.

**Deploying the DR Solution**

**Solution Deployment Overview**

1. Make sure that application data is backed up using SnapCenter with the necessary RPO requirements.
2. Provision Cloud Volumes ONTAP with the correct instance size using BlueXP within the appropriate subscription and virtual network.
   a. Configure SnapMirror for the relevant application volumes.
   b. Update the backup policies in SnapCenter to trigger SnapMirror updates after the scheduled jobs.
3. Install the Veeam software and start replicating virtual machines to Google Cloud VMware Engine instance.
4. During a disaster event, break the SnapMirror relationship using BlueXP and trigger failover of virtual machines with Veeam.
   a. Reconnect the ISCSI LUNs and NFS mounts for the application VMs.
   b. Bring up applications online.
5. Invoke failback to the protected site by reverse resyncing SnapMirror after the primary site has been recovered.

Deployment Details

Configure CVO on Google Cloud and replicate volumes to CVO

The first step is to configure Cloud Volumes ONTAP on Google Cloud (cvo) and replicate the desired volumes to Cloud Volumes ONTAP with the desired frequencies and snapshot retentions.

For sample step-by-step instructions on setting up SnapCenter and replicating the data, Refer Setup Replication with SnapCenter

Review of SQL VM protection with SnapCenter
Configure GCVE hosts and CVO data access

Two important factors to consider when deploying the SDDC are the size of the SDDC cluster in the GCVE solution and how long to keep the SDDC in service. These two key considerations for a disaster recovery solution help reduce the overall operational costs. The SDDC can be as small as three hosts, all the way up to a multi-host cluster in a full-scale deployment.

NetApp Cloud Volume Service for NFS Datastore and Cloud Volumes ONTAP for SQL databases and log can be deployed to any VPC and GCVE should have private connection to that VPC to mount NFS datastore and have VM connect to iSCSI LUNs.

To configure GCVE SDDC, see Deploy and configure the Virtualization Environment on Google Cloud Platform (GCP). As a prerequisite, verify that the guest VMs residing on the GCVE hosts are able to consume data from Cloud Volumes ONTAP after connectivity has been established.

After Cloud Volumes ONTAP and GCVE have been configured properly, begin configuring Veeam to automate the recovery of on-premises workloads to GCVE (VMs with application VMDKs and VMs with in-guest storage) by using the Veeam Replication feature and by leveraging SnapMirror for application volumes copies to Cloud Volumes ONTAP.

Install Veeam Components

Based on deployment scenario, the Veeam backup server, backup repository and backup proxy that needs to be deployed. For this use case, there is no need to deploy object store for Veeam and Scale-out repository also not required. Refer to the Veeam documentation for the installation procedure For additional information, please refer Migration with Veeam Replication

Setup VM Replication with Veeam

Both on-premises vCenter and GCVE vCenter needs to be registered with Veeam. Setup vSphere VM Replication Job At the Guest Processing step of wizard, select disable application processing as we will be utilizing SnapCenter for application aware backup and recovery.

https://netapp.hosted.panopto.com/Panopto/Pages/Embed.aspx?id=8b7e4a9b-7de1-4d48-a8e2-b01200f00692

Failover of Microsoft SQL Server VM

https://netapp.hosted.panopto.com/Panopto/Pages/Embed.aspx?id=9762dc99-081b-41a2-ac68-b01200f00ac0

Benefits of this solution

- Uses the efficient and resilient replication of SnapMirror.
- Recovers to any available points in time with ONTAP snapshot retention.
- Full automation is available for all required steps to recover hundreds to thousands of VMs, from the storage, compute, network, and application validation steps.
- SnapCenter uses cloning mechanisms that do not change the replicated volume.
- This avoids the risk of data corruption for volumes and snapshots.
- Avoids replication interruptions during DR test workflows.
- Leverages the DR data for workflows beyond DR, such as dev/test, security testing, patch and upgrade testing, and remediation testing.

- Veeam Replication allows changing VM IP addresses on DR site.

Application Disaster Recovery with SnapCenter, Cloud Volumes ONTAP and Veeam Replication

Authors: Suresh Thoppay, NetApp

Overview

Disaster recovery to cloud is a resilient and cost-effective way of protecting workloads against site outages and data corruption events such as ransomware. With NetApp SnapMirror, on-premises VMware workloads that use guest-connected storage can be replicated to NetApp Cloud Volumes ONTAP running in Google Cloud. This covers application data; however, what about the actual VMs themselves. Disaster recovery should cover all dependent components, including virtual machines, VMDKs, application data, and more. To accomplish this, SnapMirror along with Veeam can be used to seamlessly recover workloads replicated from on-premises to Cloud Volumes ONTAP while using vSAN storage for VM VMDKs.

This document provides a step-by-step approach for setting up and performing disaster recovery that uses NetApp SnapMirror, Veeam, and the Google Cloud VMware Engine (GCVE).

Assumptions

This document focuses on in-guest storage for application data (also known as guest connected), and we assume that the on-premises environment is using SnapCenter for application-consistent backups.

This document applies to any third-party backup or recovery solution. Depending on the solution used in the environment, follow best practices to create backup policies that meet organizational SLAs.
For connectivity between the on-premises environment and the Google Cloud network, use the connectivity options like dedicated interconnect or Cloud VPN. Segments should be created based on the on-premises VLAN design.

There are multiple options for connecting on-premises datacenters to Google Cloud, which prevents us from outlining a specific workflow in this document. Refer to the Google Cloud documentation for the appropriate on-premises-to-Google connectivity method.

Deploying the DR Solution

Solution Deployment Overview

1. Make sure that application data is backed up using SnapCenter with the necessary RPO requirements.
2. Provision Cloud Volumes ONTAP with the correct instance size using Cloud manager within the appropriate subscription and virtual network.
   a. Configure SnapMirror for the relevant application volumes.
   b. Update the backup policies in SnapCenter to trigger SnapMirror updates after the scheduled jobs.
3. Install the Veeam software and start replicating virtual machines to Google Cloud VMware Engine instance.
4. During a disaster event, break the SnapMirror relationship using Cloud Manager and trigger failover of virtual machines with Veeam.
   a. Reconnect the ISCSI LUNs and NFS mounts for the application VMs.
   b. Bring up applications online.
5. Invoke failback to the protected site by reverse resyncing SnapMirror after the primary site has been recovered.

Deployment Details
Configure CVO on Google Cloud and replicate volumes to CVO

The first step is to configure Cloud Volumes ONTAP on Google Cloud (cvo) and replicate the desired volumes to Cloud Volumes ONTAP with the desired frequencies and snapshot retentions.

For sample step-by-step instructions on setting up SnapCenter and replicating the data, Refer to Setup Replication with SnapCenter

Setup Replication with SnapCenter

Configure GCVE hosts and CVO data access

Two important factors to consider when deploying the SDDC are the size of the SDDC cluster in the GCVE solution and how long to keep the SDDC in service. These two key considerations for a disaster recovery solution help reduce the overall operational costs. The SDDC can be as small as three hosts, all the way up to a multi-host cluster in a full-scale deployment.

Cloud Volumes ONTAP can be deployed to any VPC and GCVE should have private connection to that VPC to have VM connect to iSCSI LUNs.

To configure GCVE SDDC, see Deploy and configure the Virtualization Environment on Google Cloud Platform (GCP). As a prerequisite, verify that the guest VMs residing on the GCVE hosts are able to consume data from Cloud Volumes ONTAP after connectivity has been established.

After Cloud Volumes ONTAP and GCVE have been configured properly, begin configuring Veeam to automate the recovery of on-premises workloads to GCVE (VMs with application VMDKs and VMs with in-guest storage) by using the Veeam Replication feature and by leveraging SnapMirror for application volumes copies to Cloud Volumes ONTAP.
Install Veeam Components

Based on deployment scenario, the Veeam backup server, backup repository and backup proxy that needs to be deployed. For this use case, there is no need to deploy object store for Veeam and Scale-out repository also not required. Refer to the Veeam documentation for the installation procedure

Setup VM Replication with Veeam

Both on-premises vCenter and GCVE vCenter needs to be registered with Veeam. Setup vSphere VM Replication Job At the Guest Processing step of wizard, select disable application processing as we will be utilizing SnapCenter for application aware backup and recovery.

Setup vSphere VM Replication Job

Failover of Microsoft SQL Server VM

Benefits of this solution

• Uses the efficient and resilient replication of SnapMirror.
• Recovers to any available points in time with ONTAP snapshot retention.
• Full automation is available for all required steps to recover hundreds to thousands of VMs, from the storage, compute, network, and application validation steps.
• SnapCenter uses cloning mechanisms that do not change the replicated volume.
  ◦ This avoids the risk of data corruption for volumes and snapshots.
  ◦ Avoids replication interruptions during DR test workflows.
  ◦ Leverages the DR data for workflows beyond DR, such as dev/test, security testing, patch and upgrade testing, and remediation testing.
• Veeam Replication allows changing VM IP addresses on DR site.

Migrating Workloads on GCP / GCVE

Migrate workloads to NetApp Cloud Volume Service datastore on Google Cloud VMware Engine using VMware HCX - Quickstart guide

Author(s): NetApp Solutions Engineering

Overview: Migrating virtual machines with VMware HCX, NetApp Cloud Volume Service datastores, and Google Cloud VMware Engine (GCVE)

One of the most common use cases for the Google Cloud VMware Engine and Cloud Volume Service datastore is the migration of VMware workloads. VMware HCX is a preferred option and provides various migration mechanisms to move on-premises virtual machines (VMs) and its data to Cloud Volume Service NFS datastores.

VMware HCX is primarily a migration platform that is designed to simplify application migration, workload
rebalancing, and even business continuity across clouds. It is included as part of Google Cloud VMware Engine Private Cloud and offers many ways to migrate workloads and can be used for disaster recovery (DR) operations.

This document provides step-by-step guidance for provisioning Cloud Volume Service datastore followed by downloading, deploying, and configuring VMware HCX, including all its main components in on-premises and the Google Cloud VMware Engine side including Interconnect, Network Extension, and WAN optimization for enabling various VM migration mechanisms.

VMware HCX works with any datastore type as the migration is at the VM level. Hence this document is applicable to existing NetApp customers and non-NetApp customers who are planning to deploy Cloud Volume Service with Google Cloud VMware Engine for a cost-effective VMware cloud deployment.

**High-level steps**

This list provides the high-level steps necessary to pair & Migrate the VMs to HCX Cloud Manager on the Google Cloud VMware Engine side from HCX Connector on-premises:

1. Prepare HCX through the Google VMware Engine portal.
2. Download and deploy the HCX Connector Open Virtualization Appliance (OVA) installer in the on-premises VMware vCenter Server.
3. Activate HCX with the license key.
4. Pair the on-premises VMware HCX Connector with Google Cloud VMware Engine HCX Cloud Manager.
5. Configure the network profile, compute profile, and service mesh.
6. (Optional) Perform network extension to avoid re-IP during migrations.
7. Validate the appliance status and ensure that migration is possible.
8. Migrate the VM workloads.
Prerequisites

Before you begin, make sure the following prerequisites are met. For more information, see this link. After the prerequisites, including connectivity, are in place, download HCX license key from the Google Cloud VMware Engine portal. After the OVA installer is downloaded, proceed with the installation process as described below.

HCX advanced is the default option and VMware HCX Enterprise edition is also available through a support ticket and supported at no additional cost. Refer this link

- Use an existing Google Cloud VMware Engine software-defined data center (SDDC) or create a private cloud by using this NetApp link or this Google link.
- Migration of VMs and associated data from the on-premises VMware vSphere-enabled data center requires network connectivity from the data center to the SDDC environment. Before migrating workloads, set up a Cloud VPN or Cloud Interconnect connection between the on-premises environment and the respective private cloud.
- The network path from on-premises VMware vCenter Server environment to the Google Cloud VMware Engine private cloud must support the migration of VMs by using vMotion.
- Make sure the required firewall rules and ports are allowed for vMotion traffic between the on-premises vCenter Server and SDDC vCenter.
- Cloud Volume Service NFS volume should be mounted as a datastore in Google Cloud VMware Engine. Follow the steps detailed in this link to attach Cloud Volume Service datastores to Google Cloud VMware Engines hosts.
High Level Architecture

For testing purposes, the lab environment from on-premises used for this validation was connected through a Cloud VPN, which allows on-premises connectivity to Google Cloud VPC.

For more detailed diagram on HCX, please refer VMware link

Solution Deployment

Follow the series of steps to complete the deployment of this solution:
Step 1: Prepare HCX through the Google VMware Engine Portal

HCX Cloud Manager component automatically gets installed as you provision private cloud with VMware Engine. To prepare for site pairing, complete the following steps:

1. Log in to the Google VMware Engine Portal and sign-in to the HCX Cloud Manager.

You can login to HCX Console either by clicking on the HCX version link

or clicking on HCX FQDN under vSphere Management Network tab.

2. In HCX Cloud Manager, go to Administration > System Updates.

3. Click Request download link and download the OVA file.
4. Update HCX Cloud Manager to the latest version available from the HCX Cloud Manager UI.
Step 2: Deploy the installer OVA in the on-premises vCenter Server

For the on-premises Connector to connect to the HCX Manager in Google Cloud VMware Engine, make sure the appropriate firewall ports are open in the on-premises environment.

To download and install HCX Connector in the on-premises vCenter Server, complete the following steps:

1. Have the ova downloaded from the HCX Console on Google Cloud VMware Engine as stated in previous step.

2. After the OVA is downloaded, deploy it on to the on-premises VMware vSphere environment by using the Deploy OVF Template option.

3. Enter all the required information for the OVA deployment, click Next, and then click Finish to deploy the VMware HCX connector OVA.

   ![](image)

   Power on the virtual appliance manually.

For step-by-step instructions, see the VMware HCX User Guide.
Step 3: Activate HCX Connector with the license key

After you deploy the VMware HCX Connector OVA on-premises and start the appliance, complete the following steps to activate HCX Connector. Generate the license key from the Google Cloud VMware Engine portal and activate it in VMware HCX Manager.

1. From the VMware Engine portal, Click on Resources, select the private cloud, and click on download icon under HCX Manager Cloud Version.

   ![Google Cloud VMware Engine](image)

   Open Downloaded file and copy the License Key String.

2. Log into the on-premises VMware HCX Manager at https://hcxmanagerIP:9443 using administrator credentials.

   Use the hcxmanagerIP and password defined during the OVA deployment.

3. In the licensing, enter the key copied from step 3 and click Activate.

   The on-premises HCX Connector should have internet access.

4. Under Datacenter Location, provide the nearest location for installing the VMware HCX Manager on-premises. Click Continue.

5. Under System Name, update the name and click Continue.

6. Click Yes, Continue.

7. Under Connect your vCenter, provide the fully qualified domain name (FQDN) or IP address of vCenter Server and the appropriate credentials and click Continue.

   Use the FQDN to avoid connectivity issues later.

8. Under Configure SSO/PSC, provide the Platform Services Controller’s(PSC) FQDN or IP address and click Continue.

   For Embedded PSC, Enter the VMware vCenter Server FQDN or IP address.

9. Verify that the information entered is correct and click Restart.

10. After the services restart, vCenter Server is displayed as green on the page that appears. Both vCenter Server and SSO must have the appropriate configuration parameters, which should be the
This process should take approximately 10 to 20 minutes and for the plug-in to be added to the vCenter Server.
Step 4: Pair on-premises VMware HCX Connector with Google Cloud VMware Engine HCX Cloud Manager

After HCX Connector is deployed and configured on on-premises vCenter, establish connection to Cloud Manager by adding the pairing. To configure the site pairing, complete the following steps:

1. To create a site pair between the on-premises vCenter environment and Google Cloud VMware Engine SDDC, log in to the on-premises vCenter Server and access the new HCX vSphere Web Client plug-in.

2. Under Infrastructure, click **Add a Site Pairing**.

   Enter the Google Cloud VMware Engine HCX Cloud Manager URL or IP address and the credentials for user with Cloud-Owner-Role privileges for accessing the private cloud.
3. Click **Connect**.

VMware HCX Connector must be able to route to HCX Cloud Manager IP over port 443.

4. After the pairing is created, the newly configured site pairing is available on the HCX Dashboard.
Step 5: Configure the network profile, compute profile, and service mesh

The VMware HCX Interconnect service appliance provides replication and vMotion-based migration capabilities over the internet and private connections to the target site. The interconnect provides encryption, traffic engineering, and VM mobility. To create an Interconnect service appliance, complete the followings steps:

1. Under Infrastructure, select **Interconnect > Multi-Site Service Mesh > Compute Profiles > Create Compute Profile**.

   The compute profiles define the deployment parameters including the appliances that are deployed and which portion of the VMware data center are accessible to HCX service.

2. After the compute profile is created, create the network profiles by selecting **Multi-Site Service Mesh > Network Profiles > Create Network Profile**.

   The network profile defines a range of IP address and networks that are used by HCX for its virtual appliances.

   This step requires two or more IP addresses. These IP addresses are assigned from the management network to the Interconnect Appliances.
3. At this time, the compute and network profiles have been successfully created.

4. Create the Service Mesh by selecting the **Service Mesh** tab within the **Interconnect** option and select the on-premises and GCVE SDDC sites.

5. The Service Mesh specifies a local and remote compute and network profile pair.

   As part of this process, the HCX appliances are deployed and automatically configured on both the source and target sites in order to create a secure transport fabric.
6. This is the final step of configuration. This should take close to 30 minutes to complete the deployment. After the service mesh is configured, the environment is ready with the IPsec tunnels successfully created to migrate the workload VMs.
Step 6: Migrate workloads

Workloads can be migrated bidirectionally between on-premises and GCVE SDDCs using various VMware HCX migration technologies. VMs can be moved to and from VMware HCX-activated entities using multiple migration technologies such as HCX bulk migration, HCX vMotion, HCX Cold migration, HCX Replication Assisted vMotion (available with HCX Enterprise edition), and HCX OS Assisted Migration (available with the HCX Enterprise edition).

To learn more about various HCX migration mechanisms, see VMware HCX Migration Types.

The HCX-IX appliance uses the Mobility Agent service to perform vMotion, Cold, and Replication Assisted vMotion (RAV) migrations.

The HCX-IX appliance adds the Mobility Agent service as a host object in the vCenter Server. The processor, memory, storage and networking resources displayed on this object do not represent actual consumption on the physical hypervisor hosting the IX appliance.

HCX vMotion

This section describes the HCX vMotion mechanism. This migration technology uses the VMware vMotion protocol to migrate a VM to GCVE. The vMotion migration option is used for migrating the VM state of a single VM at a time. There is no service interruption during this migration method.

Network Extension should be in place (for the port group in which the VM is attached) in order to migrate the VM without the need to make an IP address change.

1. From the on-premises vSphere client, go to Inventory, right-click on the VM to be migrated, and select HCX Actions > Migrate to HCX Target Site.
2. In the Migrate Virtual Machine wizard, select the Remote Site Connection (target GCVE).

3. Update the mandatory fields (Cluster, Storage, and Destination Network), Click Validate.

4. After the validation checks are complete, click Go to initiate the migration.
The vMotion transfer captures the VM active memory, its execution state, its IP address, and its MAC address. For more information about the requirements and limitations of HCX vMotion, see Understanding VMware HCX vMotion and Cold Migration.

5. You can monitor the progress and completion of the vMotion from the HCX > Migration dashboard.

The target CVS NFS datastore should have sufficient space to handle the migration.

Conclusion

Whether you’re targeting all-cloud or hybrid cloud and data residing on any type/vendor storage in on-premises, Cloud Volume Service and HCX provide excellent options to deploy and migrate the application workloads while reducing the TCO by making the data requirements seamless to the application layer. Whatever the use case, choose Google Cloud VMware Engine along with Cloud Volume Service for rapid realization of cloud benefits, consistent infrastructure, and operations across on-premises and multiple clouds, bidirectional portability of workloads, and enterprise-grade capacity and performance. It is the same familiar process and procedures used to connect the storage and migrate VMs using VMware vSphere Replication, VMware vMotion, or even network file copy (NFC).

Takeaways

The key points of this document include:

- You can now use Cloud Volume Service as a datastore on Google Cloud VMware Engine SDDC.
- You can easily migrate data from on-premises to Cloud Volume Service datastore.
- You can easily grow and shrink the Cloud Volume Service datastore to meet the capacity and performance requirements during migration activity.
Videos from Google and VMware for reference

From Google

- Deploy HCX Connector with GCVE
- Configure HCX ServiceMesh with GCVE
- Migrate VM with HCX to GCVE

From VMware

- HCX Connector deployment for GCVE
- HCX ServiceMesh configuration for GCVE
- HCX Workload Migration to GCVE

Where to find additional information

To learn more about the information described in this document, refer to the following website links:

- Google Cloud VMware Engine documentation
  https://cloud.google.com/vmware-engine/docs/overview
- Cloud Volume Service documentation
  https://cloud.google.com/architecture/partners/netapp-cloud-volumes
- VMware HCX User Guide

VM Migration to NetApp Cloud Volume Service NFS Datastore on Google Cloud VMware Engine using Veeam Replication feature

Overview

Authors: Suresh Thoppay, NetApp

VM Workloads running on VMware vSphere can be migrated to Google Cloud VMware Engine (GCVE) utilizing Veeam Replication feature.

This document provides a step-by-step approach for setting up and performing VM migration that uses NetApp Cloud Volume Service, Veeam, and the Google Cloud VMware Engine (GCVE).
Assumptions

This document assumes you have either Google Cloud VPN or Cloud Interconnect or other networking option in place to establish network connectivity from existing vSphere servers to Google Cloud VMware Engine.

There are multiple options for connecting on-premises datacenters to Google Cloud, which prevents us from outlining a specific workflow in this document. Refer to the Google Cloud documentation for the appropriate on-premises-to-Google connectivity method.

Deploying the Migration Solution

Solution Deployment Overview

1. Make sure NFS datastore from NetApp Cloud Volume Service is mounted on GCVE vCenter.
2. Ensure Veeam Backup Recovery is deployed on existing VMware vSphere environment
3. Create Replication Job to start replicating virtual machines to Google Cloud VMware Engine instance.
4. Perform Failover of Veeam Replication Job.
5. Perform Permanent Failover on Veeam.

Deployment Details

Make sure NFS datastore from NetApp Cloud Volume Service is mounted on GCVE vCenter

Login to GCVE vCenter and ensure NFS datastore with sufficient space is available.
If not, Please refer Mount NetApp CVS as NFS datastore on GCVE

Ensure Veeam Backup Recovery is deployed on existing VMware vSphere environment

Please refer Veeam Replication Components documentation to install required components.
Create Replication Job to start replicating virtual machines to Google Cloud VMware Engine instance.

Both on-premises vCenter and GCVE vCenter needs to be registered with Veeam. Setup vSphere VM Replication Job
Here is a video explaining how to Configure Replication Job.

Replica VM can have different IP from the source VM and can also be connected to different port group. For more details, check the video above.

Perform Failover of Veeam Replication Job

To Migrate VMs, perform Perform Failover

Perform Permanent Failover on Veeam.

To treat GCVE as your new source environment, perform Permanent Failover

Benefits of this solution

- Existing Veeam backup infrastructure can be utilized for migration.
- Veeam Replication allows changing VM IP addresses on target site.
- Has ability to remap existing data replicated outside of Veeam (like replicated data from BlueXP)
- Has ability to specify different network portgroup on target site.
- Can specify the order of VMs to power on.
- Utilizes VMware Change Block Tracking to minimize the amount of data to send across WAN.
- Capability to execute pre and post scripts for replication.
- Capability to execute pre and post scripts for snapshots.

Region Availability – Supplemental NFS datastore for Google Cloud Platform (GCP)

Unresolved directive in ehc/gcp-regions.adoc - include::../../_include/gcp-region-support.adoc

Security overview - NetApp Cloud Volumes Service (CVS) in Google Cloud

TR-4918: Security overview - NetApp Cloud Volumes Service in Google Cloud

Oliver Krause, Justin Parisi, NetApp

Document scope

Security, particularly in the cloud where infrastructure is outside of the control of storage administrators, is paramount to trusting your data to service offerings provided by cloud providers. This document is an overview of the security offerings that NetApp Cloud Volumes Service provides in Google Cloud.

Intended audience

This document’s intended audience includes, but is not limited to, the following roles:
Cloud providers
- Storage administrators
- Storage architects
- Field resources
- Business decision makers

If you have questions about the content of this technical report, see the section “Contact us.”

### Abbreviation | Definition
---|---
CVS-SW | Cloud Volumes Service, Service Type CVS
CVS-Performance | Cloud Volume Service, Service Type CVS-Performance
PSA | How Cloud Volumes Service in Google Cloud secures your data

Cloud Volumes Service in Google Cloud provides a multitude of ways to natively secure your data.

**Secure architecture and tenancy model**

Cloud Volumes Service provides a secure architecture in Google Cloud by segmenting the service management (control plane) and the data access (data plane) across different endpoints so that neither can impact the other (see the section “Cloud Volumes Service architecture”). It uses Google’s private services access (PSA) framework to provide the service. This framework distinguishes between the service producer, which is provided and operated by NetApp, and the service consumer, which is a Virtual Private Cloud (VPC) in a customer project, hosting the clients that want to access Cloud Volumes Service file shares.

In this architecture, tenants (see the section “Tenancy model”) are defined as Google Cloud projects that are completely isolated from each other unless explicitly connected by the user. Tenants allow complete isolation of data volumes, external name services, and other essential pieces of the solution from other tenants using the Cloud Volumes Service volume platform. Because the Cloud Volumes Service platform is connected through VPC peering, that isolation applies to it also. You can enable sharing of Cloud Volumes Service volumes between multiple projects by using a shared-VPC (see the section “Shared VPCs”). You can apply access controls to SMB shares and NFS exports to limit who or what can view or modify datasets.

**Strong identity management for the control plane**

In the control plane where Cloud Volumes Service configuration takes place, identity management is managed by using Identity Access Management (IAM). IAM is a standard service that enables you to control authentication (logins) and authorization (permissions) to Google Cloud project instances. All configuration is performed with Cloud Volumes Service APIs over a secure HTTPS transport using TLS 1.2 encryption, and authentication is performed by using JWT tokens for added security. The Google console UI for Cloud Volumes Service translates user input into Cloud Volumes Service API calls.

**Security hardening - Limiting attack surfaces**

Part of effective security is limiting the number of attack surfaces available in a service. Attack surfaces can include a variety of things, including data at-rest, in-flight transfers, logins, and the datasets themselves.
A managed service removes some of the attack surfaces inherently in its design. Infrastructure management, as described in the section “Service operation,” is handled by a dedicated team and is automated to reduce the number of times a human actually touches configurations, which helps reduce the number of intentional and unintentional errors. Networking is fenced off so that only necessary services can access one another. Encryption is baked into the data storage and only the data plane needs security attention from Cloud Volumes Service administrators. By hiding most of the management behind an API interface, security is achieved by limiting the attack surfaces.

Zero Trust model

Historically, IT security philosophy has been to trust but verify, and manifested as relying solely on external mechanisms (such as firewalls and intrusion detection systems) to mitigate threats. However, attacks and breaches evolved to bypass the verification in environments through phishing, social engineering, insider threats and other methods that provide the verification to enter networks and wreak havoc.

Zero Trust has become a new methodology in security, with the current mantra being “trust nothing while still verifying everything.” Therefore, nothing is allowed access by default. This mantra is enforced in a variety of ways, including standard firewalls and intrusion detection systems (IDS) and also with the following methods:

- Strong authentication methods (such as AES-encrypted Kerberos or JWT tokens)
- Single strong sources of identities (such as Windows Active Directory, Lightweight Directory Access Protocol (LDAP), and Google IAM)
- Network segmentation and secure multitenancy (only tenants are allowed access by default)
- Granular access controls with Least Privileged Access policies
- Small exclusive lists of dedicated, trusted administrators with digital audit and paper trails

Cloud Volumes Service running in Google Cloud adheres to the Zero Trust model by implementing the "trust nothing, verify everything" stance.

Encryption

Encrypt data at-rest (see the section “Data encryption at rest”) by using XTS-AES-256 ciphers with NetApp Volume Encryption (NVE) and in-flight with “SMB encryption” or NFS Kerberos 5p support. Rest easy knowing cross-region replication transfers are protected by TLS 1.2 encryption (see the section “Cross-region replication”). In addition, Google networking also provides encrypted communications (see the section “Data encryption in transit”) for an added layer of protection against attacks. For more information about transport encryption, see the section “Google Cloud network”.

Data protection and backups

Security isn’t just about the prevention of attacks. It is also about how we recover from attacks if or when they occur. This strategy includes data protection and backups. Cloud Volumes Service provides methods to replicate to other regions in case of outages (see the section “Cross-region replication”) or if a dataset is affected by a ransomware attack. It can also perform asynchronous backups of data to locations outside of the Cloud Volumes Service instance by using Cloud Volumes Service backup. With regular backups, mitigation of security events can take less time and save money and angst for administrators.

Fast ransomware mitigation with industry leading Snapshot copies

In addition to data protection and backups, Cloud Volumes Service provides support for immutable Snapshot copies (see the section “Immutable Snapshot copies”) of volumes that allow recovery from ransomware attacks (see the section “Service operation”) within seconds of discovering the issue and with minimal disruption. Recovery time and effects depend on the Snapshot schedule, but you can create Snapshot copies that provide
as little as one-hour deltas in ransomware attacks. Snapshot copies have a negligible effect on performance and capacity usage and are a low-risk, high-reward approach to protecting your datasets.

**Security considerations and attack surfaces**

The first step in understanding how to secure your data is identifying the risks and potential attack surfaces.

These include (but are not limited to) the following:

- Administration and logins
- Data at rest
- Data in flight
- Network and firewalls
- Ransomware, malware, and viruses

Understanding attack surfaces can help you to better secure your environments. Cloud Volumes Service in Google Cloud already considers many of these topics and implements security functionality by default, without any administrative interaction.

**Ensuring secure logins**

When securing your critical infrastructure components, it is imperative to make sure that only approved users can log in and manage your environments. If bad actors breach your administrative credentials, then they have the keys to the castle and can do anything they want—change configurations, delete volumes and backups, create backdoors, or disable Snapshot schedules.

Cloud Volumes Service for Google Cloud provides protection against unauthorized administrative logins through the obfuscation of storage as a service (StaaS). Cloud Volumes Service is completely maintained by the cloud provider with no availability to login externally. All setup and configuration operations are fully automated, so a human administrator never has to interact with the systems except in very rare circumstances.

If login is required, Cloud Volumes Service in Google Cloud secures logins by maintaining a very short list of trusted administrators that have access to log in to the systems. This gatekeeping helps reduce the number of potential bad actors with access. Additionally, the Google Cloud networking hides the systems behind layers of network security and exposes only what is needed to the outside world. For information about the Google Cloud, Cloud Volumes Service architecture, see the section “Cloud Volumes Service architecture.”

**Cluster administration and upgrades**

Two areas with potential security risks include cluster administration (what happens if a bad actor has admin access) and upgrades (what happens if a software image is compromised).

**Storage administration protection**

Storage provided as a service removes the added risk of exposure to administrators by removing that access to end users outside of the cloud data center. Instead, the only configuration done is for the data access plane by customers. Each tenant manages their own volumes, and no tenant can reach other Cloud Volumes Service instances. The service is managed by automation, with a very small list of trusted administrators given access to the systems through the processes covered in the section “Service operation.”

The CVS-Performance service type offers cross-region replication as an option to provide data protection to a different region in the event of a region failure. In those cases, Cloud Volumes Service can be failed over to the
unaffected region to maintain data access.

**Service upgrades**

Updates help protect vulnerable systems. Each update provides security enhancements and bug fixes that minimize attack surfaces. Software updates are downloaded from centralized repositories and are validated before the updates are allowed to verify that official images are used and that the upgrades are not compromised by bad actors.

With Cloud Volumes Service, updates are handled by the cloud provider teams, which removes risk exposure for administrator teams by providing experts well versed in configuration and upgrades that have automated and fully tested the process. Upgrades are nondisruptive, and Cloud Volumes Service maintains the latest updates for best overall results.

For information about the administrator team that performs these service upgrades, see the section "Service operation."

**Securing data at-rest**

Data-at-rest encryption is important to protect sensitive data in the event of a disk that is stolen, returned, or repurposed. Data in Cloud Volumes Service is protected at rest by using software-based encryption.

- Google-generated keys are used for CVS-SW.
- For CVS-Performance, the per-volume keys are stored in a key manager built into Cloud Volumes Service, which uses NetApp ONTAP CryptoMod to generate AES-256 encryption keys. CryptoMod is listed on the CMVP FIPS 140-2 validated modules list. See FIPS 140-2 Cert #4144.

Starting in November 2021, preview Customer-managed Encryption (CMEK) functionality was made available for CVS-Performance. This functionality allows you to encrypt the per-volume keys with per-project, per-region master-keys that are hosted in Google Key Management Service (KMS). KMS enables you to attach external key managers.

For details about how to configure KMS for CVS-Performance, see the Cloud Volumes Service documentation.

For more information about architecture, see the section “Cloud Volumes Service architecture.”

**Securing data in-flight**

In addition to securing data at rest, you must also be able to secure data when it is in flight between the Cloud Volumes Service instance and a client or replication target. Cloud Volumes Service provides encryption for in-flight data over NAS protocols by using encryption methods such as SMB encryption using Kerberos, the signing/sealing of packets, and NFS Kerberos 5p for end-to-end encryption of data transfers.

Replication of Cloud Volumes Service volumes uses TLS 1.2, which takes advantage of AES-GCM encryption methods.

Most insecure in-flight protocols such as telnet, NDMP, and so on are disabled by default. DNS, however, is not encrypted by Cloud Volumes Service (no DNS Sec support) and should be encrypted by using external network encryption when possible. See the section “Data encryption in transit” for more information about securing data in-flight.

For information about NAS protocol encryption, see the section “NAS protocols.”
Users and groups for NAS permissions

Part of securing your data in the cloud involves proper user and group authentication, where the users accessing the data are verified as real users in the environment and the groups contain valid users. These users and groups provide initial share and export access, as well as permission validation for files and folders in the storage system.

Cloud Volumes Service uses standard Active Directory-based Windows user and group authentication for SMB shares and Windows-style permissions. The service can also leverage UNIX identity providers such as LDAP for UNIX users and groups for NFS exports, NFSv4 ID validation, Kerberos authentication, and NFSv4 ACLs.

Currently only Active Directory LDAP is supported with Cloud Volumes Service for LDAP functionality.

Detection, prevention and mitigation of ransomware, malware, and viruses

Ransomware, malware, and viruses are a persistent threat to administrators, and detection, prevention, and mitigation of those threats are always top of mind for enterprise organizations. A single ransomware event on a critical dataset can potentially cost millions of dollars, so it is beneficial to do what you can to minimize the risk.

Although Cloud Volumes Service currently doesn’t include native detection or prevention measures, such as antivirus protection or automatic ransomware detection, there are ways to quickly recover from a ransomware event by enabling regular Snapshot schedules. Snapshot copies are immutable and read only pointers to changed blocks in the file system, are near instantaneous, have minimal impact on performance, and only use up space when data is changed or deleted. You can set schedules for Snapshot copies to match your desired acceptable recovery point objective (RPO)/recovery time objective (RTO) and can keep up to 1,024 Snapshot copies per volume.

Snapshot support is included at no additional cost (beyond data storage charges for changed blocks/data retained by Snapshot copies) with Cloud Volumes Service and, in the event of a ransomware attack, can be used to roll back to a Snapshot copy before the attack occurred. Snapshot restores take just seconds to complete, and you then can get back to serving data as normal. For more information, see The NetApp Solution for Ransomware.

Preventing ransomware from affecting your business requires a multilayered approach that includes one or more of the following:

- Endpoint protection
- Protection against external threats through network firewalls
- Detection of data anomalies
- Multiple backups (onsite and offsite) of critical datasets
- Regular restore tests of backups
- Immutable read-only NetApp Snapshot copies
- Multifactor authentication for critical infrastructure
- Security audits of system logins

This list is far from exhaustive but is a good blueprint to follow when dealing with the potential of ransomware attacks. Cloud Volumes Service in Google Cloud provides several ways to protect against ransomware events and reduce their effects.
**Immutable Snapshot copies**

Cloud Volumes Service natively provides immutable read-only Snapshot copies that are taken on a customizable schedule for quick point-in-time recovery in the event of data deletion or if an entire volume has been victimized by a ransomware attack. Snapshot restores to previous good Snapshot copies are fast and minimize data loss based on the retention period of your Snapshot schedules and RTO/RPO. The performance effect with Snapshot technology is negligible.

Because Snapshot copies in Cloud Volumes Service are read-only, they cannot be infected by ransomware unless the ransomware has proliferated into the dataset unnoticed and Snapshot copies have been taken of the data infected by ransomware. This is why you must also consider ransomware detection based on data anomalies. Cloud Volumes Service does not currently provide detection natively, but you can use external monitoring software.

**Backups and restores**

Cloud Volumes Service provides standard NAS client backup capabilities (such as backups over NFS or SMB).

- CVS-Performance offers cross-region volume replication to other CVS-Performance volumes. For more information, see [volume replication](#) in the Cloud Volumes Service documentation.
- CVS-SW offers service-native volume backup/restore capabilities. For more information, see [cloud backup](#) in the Cloud Volumes Service documentation.

Volume replication provides an exact copy of the source volume for fast failover in the case of a disaster, including ransomware events.

**Cross-region replication**

CVS-Performance enables you to securely replicate volumes across Google Cloud regions for data protection and archive use cases by using TLS1.2 AES 256 GCM encryption on a NetApp-controlled backend service network using specific interfaces used for replication running on Google's network. A primary (source) volume contains the active production data and replicates to a secondary (destination) volume to provide an exact replica of the primary dataset.

Initial replication transfers all blocks, but updates only transmit the changed blocks in a primary volume. For instance, if a 1TB database that resides on a primary volume is replicated to the secondary volume, then 1TB of space is transferred on the initial replication. If that database has a few hundred rows (hypothetically, a few MB) that change between the initialization and the next update, only the blocks with the changed rows are replicated to the secondary (a few MB). This helps to make sure that the transfer times remain low and keeps replication charges down.

All permissions on files and folders are replicated to the secondary volume, but share access permissions (such as export policies and rules or SMB shares and share ACLs) must be handled separately. In the case of a site failover, the destination site should leverage the same name services and Active Directory domain connections to provide consistent handling of user and group identities and permissions. You can use a secondary volume as a failover target in the event of a disaster by breaking the replication relationship, which converts the secondary volume to read-write.

Volume replicas are read-only, which provides an immutable copy of data offsite for quick recovery of data in instances where a virus has infected data or ransomware has encrypted the primary dataset. Read-only data won’t be encrypted, but, if the primary volume is affected and replication occurs, the infected blocks also replicate. You can use older, non-affected Snapshot copies to recover, but SLAs might fall out of range of the promised RTO/RPO depending on how quickly an attack is detected.

In addition, you can prevent malicious administrative actions, such as volume deletions, Snapshot deletions, or
Snapshot schedule changes, with cross-region replication (CRR) management in Google Cloud. This is done by creating custom roles that separate volume administrators, who can delete source volumes but not break mirrors and therefore cannot delete destination volumes, from CRR administrators, who cannot perform any volume operations. See Security Considerations in the Cloud Volumes Service documentation for permissions allowed by each administrator group.

**Cloud Volumes Service backup**

Although Cloud Volumes Service provides high data durability, external events can cause data loss. In the event of a security event such as a virus or ransomware, backups and restores become critical for resumption of data access in a timely manner. An administrator might accidentally delete a Cloud Volumes Service volume. Or users simply want to retain backup versions of their data for many months and keeping the extra Snapshot copy space inside the volume becomes a cost challenge. Although Snapshot copies should be the preferred way to keep backup versions for the last few weeks to restore lost data from them, they are sitting inside the volume and are lost if the volume goes away.

For all these reasons, NetApp Cloud Volumes Service offers backup services through Cloud Volumes Service backup.

Cloud Volumes Service backup generates a copy of the volume on Google Cloud Storage (GCS). It only backs up the actual data stored within the volume, not the free space. It works as incremental forever, meaning it transfers the volume content once and from there on continues backing up changed data only. Compared to classical backup concepts with multiple full backups, it saves large amounts of backup storage, reducing cost. Because the monthly price of backup space is lower compared to a volume, it is an ideal place to keep backup versions longer.

Users can use a Cloud Volumes Service backup to restore any backup version to the same or a different volume within the same region. If the source volume is deleted, the backup data is retained and needs to be managed (for example, deleted) independently.

Cloud Volumes Service backup is built into Cloud Volumes Service as option. Users can decide which volumes to protect by activating Cloud Volumes Service backup on a per-volume basis. See the Cloud Volumes Service backup documentation for information about backups, the number of maximum backup versions supported, scheduling, and pricing.

All backup data of a project is stored within a GCS bucket, which is managed by the service and not visible to the user. Each project uses a different bucket. Currently, the buckets are in same region as the Cloud Volumes Service volumes, but more options are being discussed. Consult the documentation for the latest status.

Data transport from a Cloud Volumes Service bucket to GCS uses service-internal Google networks with HTTPS and TLS1.2. Data is encrypted at-rest with Google-managed keys.

To manage Cloud Volumes Service backup (creating, deleting, and restoring backups), a user must have the roles/netappcloudvolumes.admin role.

**Architecture**

**Overview**

Part of trusting a cloud solution is understanding the architecture and how it is secured. This section calls out different aspects of the Cloud Volumes Service architecture in Google to help alleviate potential concerns about how data is secured, as well as call out areas where additional configuration steps might be required to obtain the most secure deployment.
The general architecture of Cloud Volumes Service can be broken down into two main components: the control plane and the data plane.

**Control plane**

The control plane in Cloud Volumes Service is the backend infrastructure managed by Cloud Volumes Service administrators and NetApp native automation software. This plane is completely transparent to end users and includes networking, storage hardware, software updates, and so on to help deliver value to a cloud-resident solution such as Cloud Volumes Service.

**Data plane**

The data plane in Cloud Volumes Service includes the actual data volumes and the overall Cloud Volumes Service configuration (such as access control, Kerberos authentication, and so on). The data plane is entirely under the control of the end users and the consumers of the Cloud Volumes Service platform.

There are distinct differences in how each plane is secured and managed. The following sections cover these differences, starting with a Cloud Volumes Service architecture overview.

**Cloud Volumes Service architecture**

In a manner similar to other Google Cloud native services such as CloudSQL, Google Cloud VMware Engine (GCVE), and FileStore, Cloud Volumes Service uses Google PSA to deliver the service. In PSA, services are built inside a service producer project, which uses VPC network peering to connect to the service consumer. The service producer is provided and operated by NetApp, and the service consumer is a VPC in a customer project, hosting the clients that want to access Cloud Volumes Service file shares.

The following figure, referenced from the architecture section of the Cloud Volumes Service documentation, shows a high-level view.
The part above the dotted line shows the control plane of the service, which controls the volume lifecycle. The part below the dotted line shows the data plane. The left blue box depicts the user VPC (service consumer), the right blue box is the service producer provided by NetApp. Both are connected through VPC peering.

**Tenancy model**

In Cloud Volumes Service, individual projects are considered unique tenants. This means that manipulation of volumes, Snapshot copies, and so on are performed on a per-project basis. In other words, all volumes are owned by the project that they were created in and only that project can manage and access the data inside of them by default. This is considered the control plane view of the service.

**Shared VPCs**

On the data plane view, Cloud Volumes Service can connect to a shared VPC. You can create volumes in the hosting project or in one of the service projects connected to the shared VPC. All projects (host or service) connected to that shared VPC are able to reach the volumes at the network layer (TCP/IP). Because all clients with network connectivity on the shared-VPC can potentially access the data through NAS protocols, access control on the individual volume (such as user/group access control lists (ACLs) and hostnames/IP addresses for NFS exports) must be used to control who can access the data.

You can connect Cloud Volumes Service to up to five VPCs per customer project. On the control plane, the project enables you to manage all created volumes, no matter which VPC they are connected to. On the data plane, VPCs are isolated from one another, and each volume can only be connected to one VPC.

Access to the individual volumes is controlled by protocol specific (NFS/SMB) access control mechanisms.

In other words, on the network layer, all projects connected to the shared VPC are able to see the volume, while, on the management side, the control plane only allows the owner project to see the volume.

**VPC Service Controls**

VPC Service Controls establish an access control perimeter around Google Cloud services that are attached to the internet and are accessible worldwide. These services provide access control through user identities but cannot restrict which network location requests originate from. VPC Service Controls close that gap by introducing the capabilities to restrict access to defined networks.

The Cloud Volumes Service data plane is not connected to the external internet but to private VPCs with well-defined network boundaries (perimeters). Within that network, each volume uses protocol-specific access control. Any external network connectivity is explicitly created by Google Cloud project administrators. The control plane, however, does not provide the same protections as the data plane and can be accessed by anyone from anywhere with valid credentials (JWT tokens).

In short, the Cloud Volumes Service data plane provides the capability of network access control, without the requirement to support VPC Service Controls and does not explicitly use VPC Service Controls.

**Packet sniffing/trace considerations**

Packet captures can be useful for troubleshooting network issues or other problems (such as NAS permissions, LDAP connectivity, and so on), but can also be used maliciously to gain information about network IP addresses, MAC addresses, user and group names, and what level of security is being used on endpoints. Because of the way Google Cloud networking, VPCs, and firewall rules are configured, unwanted access to network packets should be difficult to obtain without user login credentials or JWT tokens into the cloud instances. Packet captures are only possible on endpoints (such as virtual machines (VMs)) and only possible on endpoints internal to the VPC unless a shared VPC and/or external network tunnel/IP forwarding is in use to explicitly allow external traffic to endpoints. There is no way to sniff traffic outside of the clients.
When shared VPCs are used, in-flight encryption with NFS Kerberos and/or SMB encryption can mask much of the information gleaned from traces. However, some traffic is still sent in plaintext, such as DNS and LDAP queries. The following figure shows a packet capture from a plaintext LDAP query originating from Cloud Volumes Service and the potential identifying information that is exposed. LDAP queries in Cloud Volumes Service currently do not support encryption or LDAP over SSL. CVS-Performance support LDAP signing, if requested by Active Directory. CVS-SW does not support LDAP signing.

unixUserPassword is queried by LDAP and is not sent in plaintext but instead in a salted hash. By default, Windows LDAP does not populate the unixUserPassword fields. This field is only required if you need to leverage Windows LDAP for interactive logins through LDAP to clients. Cloud Volumes Service does not support interactive LDAP logins to the instances.

The following figure shows a packet capture from an NFS Kerberos conversation next to a capture of NFS over AUTH_SYS. Note how the information available in a trace differs between the two and how enabling in-flight encryption offers greater overall security for NAS traffic.
VM network interfaces

One trick attackers might attempt is to add a new network interface card (NIC) to a VM in promiscuous mode (port mirroring) or enable promiscuous mode on an existing NIC in order to sniff all traffic. In Google Cloud, adding a new NIC requires a VM to be shut down entirely, which creates alerts, so attackers cannot do this unnoticed.

In addition, NICs cannot be set to promiscuous mode at all and will trigger alerts in Google Cloud.

Control plane architecture

All management actions to Cloud Volumes Service are done through API. Cloud Volumes Service management integrated into the GCP Cloud Console also uses the Cloud Volumes Service API.
Identity and Access Management

Identity and Access Management (IAM) is a standard service that enables you to control authentication (logins) and authorization (permissions) to Google Cloud project instances. Google IAM provides a full audit trail of permissions authorization and removal. Currently Cloud Volumes Service does not provide control plane auditing.

Authorization/permission overview

IAM offers built-in, granular permissions for Cloud Volumes Service. You can find a complete list of granular permissions here.

IAM also offers two predefined roles called netappcloudvolumes.admin and netappcloudvolumes.viewer. These roles can be assigned to specific users or service accounts.

Assign appropriate roles and permission to allow IAM users to manage Cloud Volumes Service.

Examples for using granular permissions include the following:

- Build a custom role with only get/list/create/update permissions so that users cannot delete volumes.
- Use a custom role with only snapshot.* permissions to create a service account that is used to build application- consistent Snapshot integration.
- Build a custom role to delegate volumereplication.* to specific users.

Service accounts

To make Cloud Volumes Service API calls through scripts or Terraform, you must create a service account with the roles/netappcloudvolumes.admin role. You can use this service account to generate the JWT tokens required to authenticate Cloud Volumes Service API requests in two different ways:

- Generate a JSON key and use Google APIs to derive a JWT token from it. This is the simplest approach, but it involves manual secrets (the JSON key) management.
- Use Service account impersonation with roles/iam.serviceAccountTokenCreator. The code (script, Terraform, and so on.) runs with Application Default Credentials and impersonates the service account to gain its permissions. This approach reflects Google security best practices.

See Creating your service account and private key in the Google cloud documentation for more information.

Cloud Volumes Service API

Cloud Volumes Service API uses a REST-based API by using HTTPS (TLSv1.2) as the underlying network transport. You can find the latest API definition here and information about how to use the API at Cloud Volumes APIs in the Google cloud documentation.

The API endpoint is operated and secured by NetApp using standard HTTPS (TLSv1.2) functionality.

JWT tokens

Authentication to the API is performed with JWT bearer tokens (RFC-7519). Valid JWT tokens must be obtained by using Google Cloud IAM authentication. This must be done by fetching a token from IAM by providing a service account JSON key.
Audit logging

Currently, no user-accessible control plane audit logs are available.

Data plane architecture

Cloud Volumes Service for Google Cloud leverages the Google Cloud private services access framework. In this framework, users can connect to the Cloud Volumes Service. This framework uses Service Networking and VPC peering constructs like other Google Cloud services, ensuring complete isolation between tenants.

For an architecture overview of Cloud Volumes Service for Google Cloud, see Architecture for Cloud Volumes Service.

User VPCs (standalone or shared) are peered to VPCs within Cloud Volumes Service managed tenant projects, which hosts the volumes.

The preceding figure shows a project (the CVS consumer project in the middle) with three VPC networks connected to Cloud Volumes Service and multiple Compute Engine VMs (GCE1-7) sharing volumes:

- VPC1 allows GCE1 to access volumes A and B.
- VPC2 allows GCE2 and GCE4 to access volume C.
- The third VPC network is a shared VPC, shared with two service projects. It allows GCE3, GCE4, GCE5, and GCE6 to access volumes D and E. Shared VPC networks are only supported for volumes of the CVS-Performance service type.

GCE7 cannot access any volume.

Data can be encrypted both in-transit (using Kerberos and/or SMB encryption) and at-rest in Cloud Volumes
Data encryption in transit

Data in transit can be encrypted at the NAS protocol layer, and the Google Cloud network itself is encrypted, as described in the following sections.

Google Cloud network

Google Cloud encrypts traffic on the network level as described in Encryption in transit in the Google documentation. As mentioned in the section “Cloud Volumes Services architecture,” Cloud Volumes Service is delivered out of a NetApp-controlled PSA producer project.

In case of CVS-SW, the producer tenant runs Google VMs to provide the service. Traffic between user VMs and Cloud Volumes Service VMs is encrypted automatically by Google.

Although the data path for CVS-Performance isn’t fully encrypted on the network layer, NetApp and Google use a combination of IEEE 802.1AE encryption (MACSec), encapsulation (data encryption), and physically restricted networks to protect data in transit between the Cloud Volumes Service CVS-Performance service type and Google Cloud.

NAS protocols

NFS and SMB NAS protocols provide optional transport encryption at the protocol layer.

SMB encryption

SMB encryption provides end-to-end encryption of SMB data and protects data from eavesdropping occurrences on untrusted networks. You can enable encryption for both the client/server data connection (only available to SMB3.x capable clients) and the server/domain controller authentication.

When SMB encryption is enabled, clients that do not support encryption cannot access the share.

Cloud Volumes Service supports RC4-HMAC, AES-128-CTS-HMAC-SHA1, and AES-256-CTS-HMAC-SHA1 security ciphers for SMB encryption. SMB negotiates to the highest supported encryption type by the server.

NFSv4.1 Kerberos

For NFSv4.1, CVS-Performance offers Kerberos authentication as described in RFC7530. You can enable Kerberos on a per-volume basis.

The current strongest available encryption type for Kerberos is AES-256-CTS-HMAC-SHA1. NetApp Cloud Volumes Service supports AES-256-CTS-HMAC-SHA1, AES-128-CTS-HMAC-SHA1, DES3, and DES for NFS. It also supports ARCFOUR-HMAC (RC4) for CIFS/SMB traffic, but not for NFS.

Kerberos provides three different security levels for NFS mounts that offer choices for how strong the Kerberos security should be.

As per RedHat's Common Mount Options documentation:
sec=krb5 uses Kerberos V5 instead of local UNIX UIDs and GIDs to authenticate users.
sec=krb5i uses Kerberos V5 for user authentication and performs integrity checking of NFS operations using secure checksums to prevent data tampering.
sec=krb5p uses Kerberos V5 for user authentication, integrity checking, and encrypts NFS traffic to prevent traffic sniffing. This is the most secure setting, but it also involves the most performance overhead.

As a general rule, the more the Kerberos security level has to do, the worse the performance is, as the client and server spend time encrypting and decrypting NFS operations for each packet sent. Many clients and NFS servers provide support for AES-NI offloading to the CPUs for a better overall experience, but the performance impact of Kerberos 5p (full end-to-end encryption) is significantly greater than the impact of Kerberos 5 (user authentication).

The following table shows differences in what each level does for security and performance.

<table>
<thead>
<tr>
<th>Security level</th>
<th>Security</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFSv3—sys</td>
<td>• Least secure; plain text with numeric user IDs/group IDs</td>
<td>• Best for most cases</td>
</tr>
<tr>
<td></td>
<td>• Able to view UID, GID, client IP addresses, export paths, file names, permission in packet captures</td>
<td></td>
</tr>
<tr>
<td>NFSv4.x—sys</td>
<td>• More secure than NFSv3 (client IDs, name string/domain string matching) but still plain text</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Able to view UID, GID, client IP addresses, name strings, domain IDs, export paths, file names, permissions in packet captures</td>
<td>• Good for sequential workloads (such as VMs, databases, large files)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Bad with high file count/high metadata (30-50% worse)</td>
</tr>
<tr>
<td>Security level</td>
<td>Security</td>
<td>Performance</td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| NFS—krb5      | • Kerberos encryption for credentials in every NFS packet—wraps UID/GID of users/groups in RPC calls in GSS wrapper  
• User requesting access to mount needs a valid Kerberos ticket (either through username/password or manual key tab exchange); ticket expires after a specified time period and user must reauthenticate for access  
• No encryption for NFS operations or ancillary protocols like mount/portmapper/nlm (can see export paths, IP addresses, file handles, permissions, file names, atime/mtime in packet captures) | • Best in most cases for Kerberos; worse than AUTH_SYS |
| NFS—krb5i     | • Kerberos encryption for credentials in every NFS packet—wraps UID/GID of users/groups in RPC calls in GSS wrapper  
• User requesting access to mount needs a valid Kerberos ticket (either via username/password or manual key tab exchange); ticket expires after a specified time period and user must reauthenticate for access  
• No encryption for NFS operations or ancillary protocols like mount/portmapper/nlm (can see export paths, IP addresses, file handles, permissions, file names, atime/mtime in packet captures)  
• Kerberos GSS checksum is added to every packet to ensure nothing intercepts the packets. If checksums match, conversation is allowed. | • Better than krb5p because the NFS payload is not encrypted; only added overhead compared to krb5 is the integrity checksum. Performance of krb5i won’t be much worse than krb5 but will see some degradation. |
<table>
<thead>
<tr>
<th>Security level</th>
<th>Security</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFS – krb5p</td>
<td>• Kerberos encryption for credentials in every NFS packet—wraps UID/GID of</td>
<td>• Worst performance of the security levels; krb5p has to encrypt/decrypt more.</td>
</tr>
<tr>
<td></td>
<td>users/groups in RPC calls in GSS wrapper</td>
<td>• Better performance than krb5p with NFSv4.x for high file count workloads.</td>
</tr>
<tr>
<td></td>
<td>• User requesting access to mount needs a valid Kerberos ticket (either</td>
<td></td>
</tr>
<tr>
<td></td>
<td>via username/password or manual keytab exchange); ticket expires after</td>
<td></td>
</tr>
<tr>
<td></td>
<td>specified time period and user must reauthenticate for access</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• All of the NFS packet payloads are encrypted with the GSS wrapper</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(cannot see file handles, permissions, file names, atime/mtime in packet</td>
<td></td>
</tr>
<tr>
<td></td>
<td>captures).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Includes integrity check.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NFS operation type is visible (FSINFO, ACCESS, GETATTR, and so on).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Ancillary protocols (mount, portmap, nlm, and so on) are not encrypted</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- (can see export paths, IP addresses)</td>
<td></td>
</tr>
</tbody>
</table>

In Cloud Volumes Service, a configured Active Directory server is used as Kerberos server and LDAP server (to lookup user identities from an RFC2307 compatible schema). No other Kerberos or LDAP servers are supported. NetApp highly recommends that you use LDAP for identity management in Cloud Volumes Service. For information on how NFS Kerberos is shown in packet captures, see the section “Packet sniffing/trace considerations.”

**Data encryption at rest**

All volumes in Cloud Volumes Service are encrypted-at-rest using AES-256 encryption, which means all user data written to media is encrypted and can only be decrypted with a per-volume key.

- For CVS-SW, Google-generated keys are used.
- For CVS-Performance, the per-volume keys are stored in a key manager built into the Cloud Volumes Service.

Starting in November 2021, preview customer-managed encryption keys (CMEK) functionality was made available. This enables you to encrypt the per-volume keys with a per-project, per-region master key that is hosted in Google Key Management Service (KMS). KMS enables you to attach external key managers.
For information about configuring KMS for CVS-Performance, see Setting up customer-managed encryption keys.

**Firewall**

Cloud Volumes Service exposes multiple TCP ports to serve NFS and SMB shares:

- Ports required for NFS access
- Ports required for SMB access

Additionally, SMB, NFS with LDAP including Kerberos, and dual-protocol configurations require access to a Windows Active Directory domain. Active Directory connections must be configured on a per-region basis. Active Directory Domain controllers (DC) are identified by using DNS-based DC discovery using the specified DNS servers. Any of the DCs returned are used. The list of eligible DCs can be limited by specifying an Active Directory site.

Cloud Volumes Service reaches out with IP addresses from the CIDR range allocated with the `gcloud compute address` command while on-boarding the Cloud Volumes Service. You can use this CIDR as source addresses to configure inbound firewalls to your Active Directory domain controllers.

Active Directory Domain Controllers must expose ports to the Cloud Volumes Service CIDRs as mentioned here.

**NAS protocols**

**NAS protocols overview**

NAS protocols include NFS (v3 and v4.1) and SMB/CIFS (2.x and 3.x). These protocols are how CVS allows shared access to data across multiple NAS clients. In addition, Cloud Volumes Service can provide access to NFS and SMB/CIFS clients simultaneously (dual-protocol) while honoring all of the identity and permission settings on files and folders in the NAS shares. To maintain the highest possible data transfer security, Cloud Volumes Service supports protocol encryption in flight using SMB encryption and NFS Kerberos 5p.

- Dual-protocol is available with CVS-Performance only.

**Basics of NAS protocols**

NAS protocols are ways for multiple clients on a network to access the same data on a storage system, such as Cloud Volumes Service on GCP. NFS and SMB are the defined NAS protocols and operate on a client/server basis where Cloud Volumes Service acts as the server. Clients send access, read, and write requests to the server, and the server is responsible for coordinating the locking mechanisms for files, storing permissions and handling identity and authentication requests.

For example, the following general process is followed if a NAS client wants to create a new file in a folder.

1. The client asks the server for information about the directory (permissions, owner, group, file ID, available space, and so on); the server responds with the information if the requesting client and user have the necessary permissions on the parent folder.
2. If the permissions on the directory allow access, the client then asks the server if the file name being created already exists in the file system. If the file name is already in use, creation fails. If the file name does not exist, the server lets the client know it can proceed.

3. The client issues a call to the server to create the file with the directory handle and file name and sets the access and modified times. The server issues a unique file ID to the file to make sure that no other files are created with the same file ID.

4. The client sends a call to check file attributes before the WRITE operation. If permissions allow it, the client then writes the new file. If locking is used by the protocol/application, the client asks the server for a lock to prevent other clients from accessing the file while locked to prevent data corruption.

NFS

NFS is a distributed file system protocol that is an open IETF standard defined in Request for Comments (RFC) that allows anyone to implement the protocol.

Volumes in Cloud Volumes Service are shared out to NFS clients by exporting a path that is accessible to a client or set of clients. Permissions to mount these exports are defined by export policies and rules, which are configurable by Cloud Volumes Service administrators.

The NetApp NFS implementation is considered a gold standard for the protocol and is used in countless enterprise NAS environments. The following sections cover NFS and specific security features available in Cloud Volumes Service and how they are implemented.

Default local UNIX users and groups

Cloud Volumes Service contains several default UNIX users and groups for various basic functionalities. These users and groups cannot currently be modified or deleted. New local users and groups cannot currently be added to Cloud Volumes Service. UNIX users and groups outside of the default users and groups need to be provided by an external LDAP name service.

The following table shows the default users and groups and their corresponding numeric IDs. NetApp recommends not creating new users or groups in LDAP or on the local clients that re-use these numeric IDs.

<table>
<thead>
<tr>
<th>Default users: numeric IDs</th>
<th>Default groups: numeric IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>• root:0</td>
<td>• root:0</td>
</tr>
<tr>
<td>• pcuser:65534</td>
<td>• daemon:1</td>
</tr>
<tr>
<td>• nobody:65535</td>
<td>• pcuser:65534</td>
</tr>
<tr>
<td></td>
<td>• nobody:65535</td>
</tr>
</tbody>
</table>

When using NFSv4.1, the root user might display as nobody when running directory listing commands on NFS clients. This is due to the client's ID domain mapping configuration. See the section called NFSv4.1 and the nobody user/group for details on this issue and how to resolve it.

The root user

In Linux, the root account has access to all commands, files, and folders in a Linux-based file system. Because of the power of this account, security best practices often require the root user to be disabled or restricted in some fashion. In NFS exports, the power a root user has over the files and folders can be controlled in Cloud Volumes Service through export policies and rules and a concept known as root squash.
Root squashing ensures that the root user accessing an NFS mount is squashed to the anonymous numeric user 65534 (see the section “The anonymous user”) and is currently only available when using CVS-Performance by selecting Off for root access during export policy rule creation. If the root user is squashed to the anonymous user, it no longer has access to run chown or setuid/setgid commands (the sticky bit) on files or folders in the NFS mount, and files or folders created by the root user show the anon UID as the owner/group. In addition, NFSv4 ACLs cannot be modified by the root user. However, the root user still has access to chmod and deleted files that it does not have explicit permissions for. If you want to limit access to a root user’s file and folder permissions, consider using a volume with NTFS ACLs, creating a Windows user named root, and applying the desired permissions to the files or folders.

The anonymous user

The anonymous (anon) user ID specifies a UNIX user ID or username that is mapped to client requests that arrive without valid NFS credentials. This can include the root user when root squashing is used. The anon user in Cloud Volumes Service is 65534.

This UID is normally associated with the username nobody or nfsnobody in Linux environments. Cloud Volumes Service also uses 65534 as the local UNIX user `pcuser` (see the section “Default local UNIX users and groups”), which is also the default fallback user for Windows to UNIX name mappings when no valid matching UNIX user can be found in LDAP.

Because of the differences in usernames across Linux and Cloud Volumes Service for UID 65534, the name string for users mapped to 65534 might not match when using NFSv4.1. As a result, you might see nobody as the user on some files and folders. See the section “NFSv4.1 and the nobody user/group” for information about this issue and how to resolve it.

Access control/exports

Initial export/share access for NFS mounts is controlled through host- based export policy rules contained within an export policy. A host IP, host name, subnet, netgroup, or domain is defined to allow access to mount the NFS share and the level of access allowed to the host. Export policy rule configuration options depend on the Cloud Volumes Service level.

For CVS-SW, the following options are available for export-policy configuration:

- **Client match.** Comma-separated list of IP addresses, comma-separated list of hostnames, subnets, netgroups, domain names.
- **RO/RW access rules.** Select read/write or read only to control level of access to export. CVS-Performance provides the following options:
  - **Client match.** Comma-separated list of IP addresses, comma-separated list of hostnames, subnets, netgroups, domain names.
  - **RO/RW access rules.** Select read/write or read only to control level of access to export.
- **Root access (on/off).** Configures root squash (see the section “The root user” for details).
- **Protocol type.** This limits access to the NFS mount to a specific protocol version. When specifying both NFSv3 and NFSv4.1 for the volume, either leave both blank or check both boxes.
- **Kerberos security level (when Enable Kerberos is selected).** Provides the options of krb5, krb5i, and/or krb5p for read-only or read-write access.

Change ownership (chown) and change group (chgrp)

NFS on Cloud Volumes Service only allows the root user to run chown/chgrp on files and folders. Other users see an Operation not permitted error— even on files they own. If you use root squash (as covered in
the section “The root user”), the root is squashed to a nonroot user and is not allowed access to chown and chgrp. There are currently no workarounds in Cloud Volumes Service to allow chown and chgrp for non-root users. If ownership changes are required, consider using dual protocol volumes and set the security style to NTFS to control permissions from the Windows side.

Permission management

Cloud Volumes Service supports both mode bits (such as 644, 777, and so on for rwx) and NFSv4.1 ACLs to control permissions on NFS clients for volumes that use the UNIX security style. Standard permission management is used for these (such as chmod, chown, or nfs4_setfacl) and work with any Linux client that supports them.

Additionally, when using dual protocol volumes set to NTFS, NFS clients can leverage Cloud Volumes Service name mapping to Windows users, which then are used to resolve the NTFS permissions. This requires an LDAP connection to Cloud Volumes Service to provide numeric-ID-to-username translations because Cloud Volumes Service requires a valid UNIX username to map properly to a Windows username.

Providing granular ACLs for NFSv3

Mode bit permissions cover only owner, group, and everyone else in the semantics—meaning that there are no granular user access controls in place for basic NFSv3. Cloud Volumes Service does not support POSIX ACLs, nor extended attributes (such as chattr), so granular ACLs are only possible in the following scenarios with NFSv3:

- NTFS security style volumes (CIFS server required) with valid UNIX to Windows user mappings.
- NFSv4.1 ACLs applied using an admin client mounting NFSv4.1 to apply ACLs.

Both methods require an LDAP connection for UNIX identity management and a valid UNIX user and group information populated (see the section “LDAP”) and are only available with CVS-Performance instances. To use NTFS security style volumes with NFS, you must use dual-protocol (SMB and NFSv3) or dual-protocol (SMB and NFSv4.1), even if no SMB connections are made. To use NFSv4.1 ACLs with NFSv3 mounts, you must select Both (NFSv3/NFSv4.1) as the protocol type.

Regular UNIX mode bits don’t provide the same level of granularity in permissions that NTFS or NFSv4.x ACLs provide. The following table compares the permission granularity between NFSv3 mode bits and NFSv4.1 ACLs. For information about NFSv4.1 ACLs, see nfs4_acl - NFSv4 Access Control Lists.
<table>
<thead>
<tr>
<th>NFSv3 mode bits</th>
<th>NFSv4.1 ACLs</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Set user ID on execution</td>
<td>Access control entry (ACE) types (Allow/Deny/Audit)</td>
</tr>
<tr>
<td>• Set group ID on execution</td>
<td>* Inheritance flags</td>
</tr>
<tr>
<td>• Save swapped text (not defined in POSIX)</td>
<td>* directory-inherit</td>
</tr>
<tr>
<td>• Read permission for owner</td>
<td>* file-inherit</td>
</tr>
<tr>
<td>• Write permission for owner</td>
<td>* no-propagate-inherit</td>
</tr>
<tr>
<td>• Execute permission for owner on a file; or look up (search) permission for owner in directory</td>
<td>* inherit-only</td>
</tr>
<tr>
<td>• Read permission for group</td>
<td>Permissions</td>
</tr>
<tr>
<td>• Write permission for group</td>
<td>* read-data (files) / list-directory (directories)</td>
</tr>
<tr>
<td>• Execute permission for group on a file; or look up (search) permission for group in directory</td>
<td>* write-data (files) / create-file (directories)</td>
</tr>
<tr>
<td>• Read permission for others</td>
<td>* append-data (files) / create-subdirectory (directories)</td>
</tr>
<tr>
<td>• Write permission for others</td>
<td>* execute (files) / change-directory (directories)</td>
</tr>
<tr>
<td>• Execute permission for others on a file; or look up (search) permission for others in directory</td>
<td>* delete</td>
</tr>
<tr>
<td></td>
<td>* delete-child</td>
</tr>
<tr>
<td></td>
<td>* read-attributes</td>
</tr>
<tr>
<td></td>
<td>* write-attributes</td>
</tr>
<tr>
<td></td>
<td>* read-named-attributes</td>
</tr>
<tr>
<td></td>
<td>* write-named-attributes</td>
</tr>
<tr>
<td></td>
<td>* read-ACL</td>
</tr>
<tr>
<td></td>
<td>* write-ACL</td>
</tr>
<tr>
<td></td>
<td>* write-owner</td>
</tr>
<tr>
<td></td>
<td>* Synchronize</td>
</tr>
</tbody>
</table>

Finally, NFS group membership (in both NFSv3 and NFSV4.x) is limited to a default maximum of 16 for AUTH_SYS as per the RPC packet limits. NFS Kerberos provides up to 32 groups and NFSv4 ACLs remove the limitation by way of granular user and group ACLs (up to 1024 entries per ACE).

Additionally, Cloud Volumes Service provides extended group support to extend the maximum supported groups up to 32. This requires an LDAP connection to an LDAP server that contains valid UNIX user and group identities. For more information about configuring this, see Creating and managing NFS volumes in the Google documentation.

**NFSv3 user and group IDs**

NFSv3 user and group IDs come across the wire as numeric IDs rather than names. Cloud Volumes Service does no username resolution for these numeric IDs with NFSv3, with UNIX security style volumes using just mode bits. When NFSv4.1 ACLs are present, a numeric ID lookup and/or name string lookup is needed to resolve the ACL properly—even when using NFSv3. With NTFS security style volumes, Cloud Volumes Service must resolve a numeric ID to a valid UNIX user and then map to a valid Windows user to negotiate access rights.

**Security limitations of NFSv3 user and group IDs**

With NFSv3, the client and server never have to confirm that the user attempting a read or write with a numeric ID is a valid user; it is just implicitly trusted. This opens the file system up to potential breaches simply by spoofing any numeric ID. To prevent security holes like this, there are a few options available to Cloud Volumes Service.

- Implementing Kerberos for NFS forces users to authenticate with a username and password or keytab file to get a Kerberos ticket to allow access into a mount. Kerberos is available with CVS-Performance instances and only with NFSv4.1.
• Limiting the list of hosts in your export policy rules limits which NFSv3 clients have access to the Cloud Volumes Service volume.

• Using dual-protocol volumes and applying NTFS ACLs to the volume forces NFSv3 clients to resolve numeric IDs to valid UNIX usernames to authenticate properly to access mounts. This requires enabling LDAP and configuring UNIX user and group identities.

• Squashing the root user limits the damage a root user can do to an NFS mount but does not completely remove risk. For more information, see the section “The root user.”

Ultimately, NFS security is limited to what the protocol version you are using offers. NFSv3, while more performant in general than NFSv4.1, does not provide the same level of security.

NFSv4.1

NFSv4.1 provides greater security and reliability as compared to NFSv3, for the following reasons:

• Integrated locking through a lease-based mechanism
• Stateful sessions
• All NFS functionality over a single port (2049)
• TCP only
• ID domain mapping
• Kerberos integration (NFSv3 can use Kerberos, but only for NFS, not for ancillary protocols such as NLM)

NFSv4.1 dependencies

Because of the additionally security features in NFSv4.1, there are some external dependencies involved that were not needed to use NFSv3 (similar to how SMB requires dependencies such as Active Directory).

NFSv4.1 ACLs

Cloud Volumes Service offers support for NFSv4.x ACLs, which deliver distinct advantages over normal POSIX-style permissions, such as the following:

• Granular control of user access to files and directories
• Better NFS security
• Improved interoperability with CIFS/SMB
• Removal of the NFS limitation of 16 groups per user with AUTH_SYS security
• ACLs bypass the need for group ID (GID) resolution, which effectively removes the GID limit

NFSv4.1 ACLs are controlled from NFS clients—not from Cloud Volumes Service. To use NFSv4.1 ACLs, be sure your client’s software version supports them and the proper NFS utilities are installed.

Compatibility between NFSv4.1 ACLs and SMB clients

NFSv4 ACLs are different from Windows file-level ACLs (NTFS ACLs) but carry similar functionality. However, in multiprotocol NAS environments, if NFSv4.1 ACLs are present and you are using dual-protocol access (NFS and SMB on the same datasets), clients using SMB2.0 and later won’t be able to view or manage ACLs from Windows security tabs.
How NFSv4.1 ACLs work

For reference, the following terms are defined:

- **Access control list (ACL).** A list of permissions entries.
- **Access control entry (ACE).** A permission entry in the list.

When a client sets an NFSv4.1 ACL on a file during a SETATTR operation, Cloud Volumes Service sets that ACL on the object, replacing any existing ACL. If there is no ACL on a file, then the mode permissions on the file are calculated from OWNER@, GROUP@, and EVERYONE@. If there are any existing SUID/SGID/STICKY bits on the file, they are not affected.

When a client gets an NFSv4.1 ACL on a file during the course of a GETATTR operation, Cloud Volumes Service reads the NFSv4.1 ACL associated with the object, constructs a list of ACEs, and returns the list to the client. If the file has an NT ACL or mode bits, then an ACL is constructed from mode bits and is returned to the client.

Access is denied if a DENY ACE is present in the ACL; access is granted if an ALLOW ACE exists. However, access is also denied if neither of the ACEs is present in the ACL.

A security descriptor consists of a security ACL (SACL) and a discretionary ACL (DACL). When NFSv4.1 interoperates with CIFS/SMB, the DACL is one-to-one mapped with NFSv4 and CIFS. The DACL consists of the ALLOW and the DENY ACEs.

If a basic `chmod` is run on a file or folder with NFSv4.1 ACLs set, existing user and group ACLs are preserved, but the default OWNER@, GROUP@, EVERYONE@ ACLs are modified.

A client using NFSv4.1 ACLs can set and view ACLs for files and directories on the system. When a new file or subdirectory is created in a directory that has an ACL, that object inherits all ACEs in the ACL that have been tagged with the appropriate **inheritance flags**.

If a file or directory has an NFSv4.1 ACL, that ACL is used to control access no matter which protocol is used to access the file or directory.

Files and directories inherit ACEs from NFSv4 ACLs on parent directories (possibly with appropriate modifications) as long as the ACEs have been tagged with the correct inheritance flags.

When a file or directory is created as the result of an NFSv4 request, the ACL on the resulting file or directory depends on whether the file creation request includes an ACL or only standard UNIX file access permissions. The ACL also depends on whether the parent directory has an ACL.

- If the request includes an ACL, that ACL is used.
- If the request includes only standard UNIX file access permissions and the parent directory does not have an ACL, the client file mode is used to set standard UNIX file access permissions.
- If the request includes only standard UNIX file access permissions and the parent directory has a noninheritable ACL, a default ACL based on the mode bits passed into the request is set on the new object.
- If the request includes only standard UNIX file access permissions but the parent directory has an ACL, the ACEs in the parent directory’s ACL are inherited by the new file or directory as long as the ACEs have been tagged with the appropriate inheritance flags.
ACE permissions

NFSv4.1 ACLs permissions uses a series of upper- and lower-case letter values (such as rxtncy) to control access. For more information about these letter values, see HOW TO: Use NFSv4 ACL.

NFSv4.1 ACL behavior with umask and ACL inheritance

NFSv4 ACLs provide the ability to offer ACL inheritance. ACL inheritance means that files or folders created beneath objects with NFSv4.1 ACLs set can inherit the ACLs based on the configuration of the ACL inheritance flag.

Umask is used to control the permission level at which files and folders are created in a directory without administrator interaction. By default, Cloud Volumes Service allows umask to override inherited ACLs, which is expected behavior as per RFC 5661.

ACL formatting

NFSv4.1 ACLs have specific formatting. The following example is an ACE set on a file:

```
A::ldapuser@domain.netapp.com:rwatTnNcCy
```

The preceding example follows the ACL format guidelines of:

```
type:flags:principal:permissions
```

A type of A means “allow.” The inherit flags are not set in this case, because the principal is not a group and does not include inheritance. Also, because the ACE is not an AUDIT entry, there is no need to set the audit flags. For more information about NFSv4.1 ACLs, see http://linux.die.net/man/5/nfs4_acl.

If the NFSv4.1 ACL is not set properly (or a name string cannot be resolved by the client and server), the ACL might not behave as expected, or the ACL change might fail to apply and throw an error.

Sample errors include:

```
Failed setxattr operation: Invalid argument
Scanning ACE string 'A:: user@rwaDxtTnNcCy' failed.
```

Explicit DENY

NFSv4.1 permissions can include explicit DENY attributes for OWNER, GROUP, and EVERYONE. That is because NFSv4.1 ACLs are default-deny, which means that if an ACL is not explicitly granted by an ACE, then it is denied. Explicit DENY attributes override any ACCESS ACEs, explicit or not.

DENY ACEs are set with an attribute tag of D.

In the example below, GROUP@ is allowed all read and execute permissions, but denied all write access.
DENY ACEs should be avoided whenever possible because they can be confusing and complicated; ALLOW ACLs that are not explicitly defined are implicitly denied. When DENY ACEs are set, users might be denied access when they expect to be granted access.

The preceding set of ACEs is equivalent to 755 in mode bits, which means:

- The owner has full rights.
- Groups have read only.
- Others have read only.

However, even if permissions are adjusted to the 775 equivalent, access can be denied because of the explicit DENY set on EVERYONE.

**NFSv4.1 ID domain mapping dependencies**

NFSv4.1 leverages ID domain mapping logic as a security layer to help verify that a user attempting access to an NFSv4.1 mount is indeed who they claim to be. In these cases, the username and group name coming from the NFSv4.1 client appends a name string and sends it to the Cloud Volumes Service instance. If that username/group name and ID string combination does not match, then the user and/or group is squashed to the default nobody user specified in the /etc/idmapd.conf file on the client.

This ID string is a requirement for proper permission adherence, especially when NFSv4.1 ACLs and/or Kerberos are in use. As a result, name service server dependencies such as LDAP servers are necessary to ensure consistency across clients and Cloud Volumes Service for proper user and group name identity resolution.

Cloud Volumes Service uses a static default ID domain name value of defaultv4iddomain.com. NFS clients default to the DNS domain name for its ID domain name settings, but you can manually adjust the ID domain name in /etc/idmapd.conf.

If LDAP is enabled in Cloud Volumes Service, then Cloud Volumes Service automates the NFS ID domain to change to what is configured for the search domain in DNS and clients won't need to be modified unless they use different DNS domain search names.

When Cloud Volumes Service can resolve a username or group name in local files or LDAP, the domain string is used and non-matching domain IDs squash to nobody. If Cloud Volumes Service cannot find a username or group name in local files or LDAP, the numeric ID value is used and the NFS client resolves the name properly (this is similar to NFSv3 behavior).

Without changing the client's NFSv4.1 ID domain to match what the Cloud Volumes Service volume is using, you see the following behavior:
UNIX users and groups with local entries in Cloud Volumes Service (such as root, as defined in local UNIX
users and groups) are squashed to the nobody value.

UNIX users and groups with entries in LDAP (if Cloud Volumes Service is configured to use LDAP)
squashes to nobody if DNS domains are different between NFS clients and Cloud Volumes Service.

UNIX users and groups with no local entries or LDAP entries use the numeric ID value and resolve to the
name specified on the NFS client. If no name exists on the client, only the numeric ID is shown.

The following shows the results of the preceding scenario:

```
# ls -la /mnt/home/prof1/nfs4/
total 8
drwxr-xr-x 2 nobody nobody 4096 Feb 3 12:07 .
drwxrwxrwx 7 root root 4096 Feb 3 12:06 ..
-rw-r--r-- 1 9835 9835 0 Feb 3 12:07 client-user-no-name
-rw-r--r-- 1 nobody nobody 0 Feb 3 12:07 ldap-user-file
-rw-r--r-- 1 nobody nobody 0 Feb 3 12:06 root-user-file
```

When the client and server ID domains match, this is how the same file listing looks:

```
# ls -la
total 8
drwxr-xr-x 2 root root 4096 Feb 3 12:07 .
drwxrwxrwx 7 root root 4096 Feb 3 12:06 ..
-rw-r--r-- 1 9835 9835 0 Feb 3 12:07 client-user-no-name
-rw-r--r-- 1 apache apache-group 0 Feb 3 12:07 ldap-user-file
-rw-r--r-- 1 root root 0 Feb 3 12:06 root-user-file
```

For more information about this issue and how to resolve it, see the section “NFSv4.1 and the nobody
user/group.”

**Kerberos dependencies**

If you plan to use Kerberos with NFS, you must have the following with Cloud Volumes Service:

- Active Directory domain for Kerberos Distribution Center services (KDC)
- Active Directory domain with user and group attributes populated with UNIX information for LDAP
  functionality (NFS Kerberos in Cloud Volumes Service requires a user SPN to UNIX user mapping for
  proper functionality.)
- LDAP enabled on the Cloud Volumes Service instance
- Active Directory domain for DNS services

**NFSv4.1 and the nobody user/group**

One of the most common issues seen with an NFSv4.1 configuration is when a file or folder is shown in a
listing using `ls` as being owned by the user:group combination of nobody:nobody.
For example:

```
sh-4.2$ ls -la | grep prof1-file
-rw-r--r-- 1 nobody nobody 0 Apr 24 13:25 prof1-file
```

And the numeric ID is 99.

```
sh-4.2$ ls -lan | grep prof1-file
-rw-r--r-- 1 99 99 0 Apr 24 13:25 prof1-file
```

In some instances, the file might show the correct owner but nobody as the group.

```
sh-4.2$ ls -la | grep newfile1
-rw-r--r-- 1 prof1 nobody 0 Oct 9 2019 newfile1
```

Who is nobody?

The nobody user in NFSv4.1 is different from the nfsnobody user. You can view how an NFS client sees each user by running the id command:

```
# id nobody
uid=99(nobody) gid=99(nobody) groups=99(nobody)
# id nfsnobody
uid=65534(nfsnobody) gid=65534(nfsnobody) groups=65534(nfsnobody)
```

With NFSv4.1, the nobody user is the default user defined by the idmapd.conf file and can be defined as any user you want to use.

```
# cat /etc/idmapd.conf | grep nobody
#Nobody-User = nobody
#Nobody-Group = nobody
```

Why does this happen?

Because security through name string mapping is a key tenet of NFSv4.1 operations, the default behavior when a name string does not match properly is to squash that user to one that won’t normally have any access to files and folders owned by users and groups.

When you see nobody for the user and/or group in file listings, this generally means something in NFSv4.1 is misconfigured. Case sensitivity can come into play here.

For example, if user1@CVSDEMO.LOCAL (uid 1234, gid 1234) is accessing an export, then Cloud Volumes Service must be able to find user1@CVSDEMO.LOCAL (uid 1234, gid 1234). If the user in Cloud Volumes Service is USER1@CVSDEMO.LOCAL, then it won’t match (uppercase USER1 versus lowercase user1). In
many cases, you can see the following in the messages file on the client:

```
May 19 13:14:29 centos7 nfssidmap[17481]: nss_getpwnam: name 'root@defaultv4iddomain.com' does not map into domain 'CVSDEMO.LOCAL'
May 19 13:15:05 centos7 nfssidmap[17534]: nss_getpwnam: name 'nobody' does not map into domain 'CVSDEMO.LOCAL'
```

The client and server must both agree that a user is indeed who they are claiming to be, so you must check the following to ensure that the user that the client sees has the same information as the user that Cloud Volumes Service sees.

- **NFSv4.x ID domain.** Client: idmapd.conf file; Cloud Volumes Service uses defaultv4iddomain.com and cannot be changed manually. If using LDAP with NFSv4.1, Cloud Volumes Service changes the ID domain to what the DNS search domain is using, which is the same as the AD domain.

- **User name and numeric IDs.** This determines where the client is looking for user names and leverages the name service switch configuration—client: nsswitch.conf and/or local passwd and group files; Cloud Volumes Service does not allow modifications to this but automatically adds LDAP to the configuration when it is enabled.

- **Group name and numeric IDs.** This determines where the client is looking for group names and leverages the name service switch configuration—client: nsswitch.conf and/or local passwd and group files; Cloud Volumes Service does not allow modifications to this but automatically adds LDAP to the configuration when it is enabled.

In almost all cases, if you see nobody in user and group listings from clients, the issue is user or group name domain ID translation between Cloud Volumes Service and the NFS client. To avoid this scenario, use LDAP to resolve user and group information between clients and Cloud Volumes Service.

**Viewing name ID strings for NFSv4.1 on clients**

If you are using NFSv4.1, there is a name-string mapping that takes place during NFS operations, as previously described.

In addition to using /var/log/messages to find an issue with NFSv4 IDs, you can use the `nfsidmap -l` command on the NFS client to view which usernames have properly mapped to the NFSv4 domain.

For example, this is output of the command after a user that can be found by the client and Cloud Volumes Service accesses an NFSv4.x mount:

```
# nfsidmap -l
4 .id_resolver keys found:
  gid:daemon@CVSDEMO.LOCAL
  uid:nfs4@CVSDEMO.LOCAL
  gid:root@CVSDEMO.LOCAL
  uid:root@CVSDEMO.LOCAL
```

When a user that does not map properly into the NFSv4.1 ID domain (in this case, netapp-user) tries to access the same mount and touches a file, they are assigned nobody:nobody, as expected.
The `nfsidmap -l` output shows the user `pcuser` in the display but not `netapp-user`; this is the anonymous user in our export-policy rule (65534).

**SMB**

SMB is a network file sharing protocol developed by Microsoft that provides centralized user/group authentication, permissions, locking, and file sharing to multiple SMB clients over an Ethernet network. Files and folders are presented to clients by way of shares, which can be configured with a variety of share properties and offers access control through share-level permissions. SMB can be presented to any client that offers support for the protocol, including Windows, Apple, and Linux clients.

Cloud Volumes Service provides support for the SMB 2.1 and 3.x versions of the protocol.

**Access control/SMB shares**

- When a Windows username requests access to the Cloud Volumes Service volume, Cloud Volumes Service looks for a UNIX username using the methods configured by Cloud Volumes Service administrators.

- If an external UNIX identity provider (LDAP) is configured and Windows/UNIX usernames are identical, then Windows usernames will map 1:1 to UNIX usernames without any additional configuration needed. When LDAP is enabled, Active Directory is used to host those UNIX attributes for user and group objects.
• If Windows names and UNIX names do not match identically, then LDAP must be configured to allow Cloud Volumes Service to use the LDAP name mapping configuration (see the section "Using LDAP for asymmetric name mapping").

• If LDAP is not in use, then Windows SMB users map to a default local UNIX user named pcuser in Cloud Volumes Service. This means files written in Windows by users that map to the pcuser show UNIX ownership as pcuser in multiprotocol NAS environments. pcuser here is effectively the nobody user in Linux environments (UID 65534).

In deployments with SMB only, the pcuser mapping still occurs, but it won’t matter, because Windows user and group ownership is correctly displayed and NFS access to the SMB-only volume is not allowed. In addition, SMB-only volumes do not support conversion to NFS or dual-protocol volumes after they are created.

Windows leverages Kerberos for username authentication with the Active Directory domain controllers, which requires a username/password exchange with the AD DCs, which is external to the Cloud Volumes Service instance. Kerberos authentication is used when the \SERVERNAME UNC path is used by the SMB clients and the following is true:

• DNS A/AAAA entry exists for SERVERNAME
• A valid SPN for SMB/CIFS access exists for SERVERNAME

When a Cloud Volumes Service SMB volume is created, the machine account name is created as defined in the section "How Cloud Volumes Service shows up in Active Directory." That machine account name also becomes the SMB share access path because Cloud Volumes Service leverages Dynamic DNS (DDNS) to create the necessary A/AAAA and PTR entries in DNS and the necessary SPN entries on the machine account principal.

For PTR entries to be created, the reverse lookup zone for the Cloud Volumes Service instance IP address must exist on the DNS server.

For example, this Cloud Volumes Service volume uses the following UNC share path: \cvs-east-433d.cvsdemo.local.

In Active Directory, these are the Cloud Volumes Service-generated SPN entries:

```
PS C:\> setspn /L CV$EAST-433D
Registered ServicePrincipalNames for CN=CV$EAST-433D, CN=Computers, DC=cv$demo, DC=local:
HOST/cvs-east-433d.cvsdemo.local
HOST/CVS-EAST-433D
```

This is the DNS forward/reverse lookup result:
Optionally, more access control can be applied by enabling/requiring SMB encryption for SMB shares in Cloud Volumes Service. If SMB encryption isn’t supported by one of the endpoints, then access is not allowed.

**Using SMB name aliases**

In some cases, it might be a security concern for end users to know the machine account name in use for Cloud Volumes Service. In other cases, you might simply want to provide a simpler access path to your end users. In those cases, you can create SMB aliases.

If you want to create aliases for the SMB share path, you can leverage what is known as a CNAME record in DNS. For example, if you want to use the name `\CIFS` to access shares instead of `\cvs-east-433d.cvsdemo.local`, but you still want to use Kerberos authentication, a CNAME in DNS that points to the existing A/AAAA record and an additional SPN added to the existing machine account provides Kerberos access.
This is the resulting DNS forward lookup result after adding a CNAME:

```
PS C:\> nslookup cifs
Server:  ok-activedirectory.us-east4-a.c.cv-solution-architect-lab.internal
Address:  10. xx.0. xx
Name:    CVS-EAST-433D.cvsdemo.local
Address:  10. xxx.0. x
Aliases: cifs.cvsdemo.local
```

This is the resulting SPN query after adding new SPNs:
In a packet capture, we can see the Session Setup Request using the SPN tied to the CNAME.

SMB authentication dialects

Cloud Volumes Service supports the following dialects for SMB authentication:

- LM
- NTLM
- NTLMv2
- Kerberos

Kerberos authentication for SMB share access is the most secure level of authentication you can use. With AES and SMB encryption enabled, the security level is further increased.

Cloud Volumes Service also supports backward compatibility for LM and NTLM authentication. When Kerberos is misconfigured (such as when creating SMB aliases), share access falls back to weaker authentication methods (such as NTLMv2). Because these mechanisms are less secure, they are disabled in some Active Directory environments. If weaker authentication methods are disabled and Kerberos is not configured properly, share access fails because there is no valid authentication method to fall back to.

For information about configuring/viewing your supported authentication levels in Active Directory, see Network security: LAN Manager authentication level.

Permission models

NTFS/File permissions

NTFS permissions are the permissions applied to files and folders in file systems adhering to NTFS logic. You can apply NTFS permissions in Basic or Advanced and can be set to Allow or Deny for access control.

Basic permissions include the following:
When you set permissions for a user or group, referred to as an ACE, it resides in an ACL. NTFS permissions use the same read/write/execute basics as UNIX mode bits, but they can also extend to more granular and extended access controls (also known as Special Permissions), such as Take Ownership, Create Folders/Append Data, Write Attributes, and more.

Standard UNIX mode bits do not provide the same level of granularity as NTFS permissions (such as being able to set permissions for individual user and group objects in an ACL or setting extended attributes). However, NFSv4.1 ACLs do provide the same functionality as NTFS ACLs.

NTFS permissions are more specific than share permissions and can be used in conjunction with share permissions. With NTFS permission structures, the most restrictive applies. As such, explicit denials to a user or group overrides even Full Control when defining access rights.

NTFS permissions are controlled from Windows SMB clients.

**Share permissions**

Share permissions are more general than NTFS permissions (Read/Change/Full Control only) and control the initial entry into an SMB share—similar to how NFS export policy rules work.

Although NFS export policy rules control access through host-based information such as IP addresses or host names, SMB share permissions can control access by using user and group ACEs in a share ACL. You can set share ACLs either from the Windows client or from the Cloud Volumes Service management UI.

By default, share ACLs and initial volume ACLs include Everyone with Full Control. The file ACLs should be changed but share permissions are overruled by the file permissions on objects in the share.

For instance, if a user is only allowed Read access to the Cloud Volumes Service volume file ACL, they are denied access to create files and folders even though the share ACL is set to Everyone with Full Control, as shown in the following figure.
For best security results, do the following:

- Remove Everyone from the share and file ACLs and instead set share access for users or groups.
- Use groups for access control instead of individual users for ease of management and faster removal/addition of users to share ACLs through group management.
- Allow less restrictive, more general share access to the ACEs on the share permissions and lock down access to users and groups with file permissions for more granular access control.
- Avoid general use of explicit deny ACLs, because they override allow ACLs. Limit use of explicit deny ACLs for users or groups that need to be restricted from access to a file system quickly.
- Make sure that you pay attention to the ACL inheritance settings when modifying permissions; setting the inheritance flag at the top level of a directory or volume with high file counts means that each file below that
directory or volume has inherited permissions added to it, which can create unwanted behavior such as unintended access/denial and long churn of permission modification as each file is adjusted.

**SMB share security features**

When you first create a volume with SMB access in Cloud Volumes Service, you are presented with a series of choices for securing that volume.

Some of these choices depend on the Cloud Volumes Service level (Performance or Software) and choices include:

- **Make snapshot directory visible (available for both CVS-Performance and CVS-SW).** This option controls whether or not SMB clients can access the Snapshot directory in an SMB share (\server\share\~snapshot and/or Previous Versions tab). The default setting is Not Checked, which means that the volume defaults to hiding and disallowing access to the ~snapshot directory, and no Snapshot copies appear in the Previous Versions tab for the volume.

**Snapshot directory visible: Unchecked**

![Snapshot directory visible: Unchecked](image1)

**Snapshot directory visible: Checked**

![Snapshot directory visible: Checked](image2)

Hiding Snapshot copies from end users might be desired for security reasons, performance reasons (hiding these folders from AV scans) or preference. Cloud Volumes Service Snapshots are read-only, so even if these Snapshots are visible, end users cannot delete or modify files in the Snapshot directory. File permissions on the files or folders at the time the Snapshot copy was taken apply. If a file or folder’s permissions change between Snapshot copies, then the changes also apply to the files or folders in the Snapshot directory. Users and groups can gain access to these files or folders based on permissions. While deletes or modifications of files in the Snapshot directory are not possible, it is possible to copy files or folders out of the Snapshot.
directory.

- **Enable SMB encryption (available for both CVS-Performance and CVS-SW)**. SMB encryption is disabled on the SMB share by default (unchecked). Checking the box enables SMB encryption, which means traffic between the SMB client and server is encrypted in-flight with the highest supported encryption levels negotiated. Cloud Volumes Service supports up to AES-256 encryption for SMB. Enabling SMB encryption does carry a performance penalty that might or might not be noticeable to your SMB clients—roughly in the 10-20% range. NetApp strongly encourages testing to see if that performance penalty is acceptable.

- **Hide SMB share (available for both CVS-Performance and CVS-SW)**. Setting this option hides the SMB share path from normal browsing. This means that clients that do not know the share path cannot see the shares when accessing the default UNC path (such as `\CVS-SMB`). When the checkbox is selected, only clients that explicitly know the SMB share path or have the share path defined by a Group Policy Object can access it (security through obfuscation).

- **Enable access-based enumeration (ABE) (CVS-SW only)**. This is similar to hiding the SMB share, except the shares or files are only hidden from users or groups that do not have permissions to access the objects. For instance, if Windows user `joe` is not allowed at least Read access through the permissions, then the Windows user `joe` cannot see the SMB share or files at all. This is disabled by default, and you can enable it by selecting the checkbox. For more information on ABE, see the NetApp Knowledge Base article [How does Access Based Enumeration (ABE) work?](#)

- **Enable Continuously Available (CA) share support (CVS-Performance only)**. Continuously Available SMB shares provide a way to minimize application disruptions during failover events by replicating lock states across nodes in the Cloud Volumes Service backend system. This is not a security feature, but it does offer better overall resiliency. Currently, only SQL Server and FSLogix applications are supported for this functionality.

**Default hidden shares**

When an SMB server is created in Cloud Volumes Service, there are hidden administrative shares (using the `$` naming convention) that are created in addition to the data volume SMB share. These include C$ (namespace access) and IPC$ (sharing named pipes for communication between programs, such as the remote procedure calls (RPC) used for Microsoft Management Console (MMC) access).

The IPC$ share contains no share ACLs and cannot be modified—it is strictly used for RPC calls and Windows disallows anonymous access to these shares by default.

The C$ share allows BUILTIN/Administrators access by default, but Cloud Volumes Service automation removes the share ACL and does not allow access to anyone because access to the C$ share allows visibility into all mounted volumes in the Cloud Volumes Service file systems. As a result, attempts to navigate to `\SERVER\C$` fail.

**Accounts with local/BUILTIN administrator/backup rights**

Cloud Volumes Service SMB servers maintain similar functionality to regular Windows SMB servers in that there are local groups (such as BUILTIN/Administrators) that apply access rights to select domain users and groups.

When you specify a user to be added to Backup Users, the user is added to the BUILTIN\Backup Operators group in the Cloud Volumes Service instance that uses that Active Directory connection, which then gets the `SeBackupPrivilege` and `SeRestorePrivilege`.

When you add a user to Security Privilege Users, the user is given the `SeSecurityPrivilege`, which is useful in some application use cases, such as SQL Server on SMB shares.
Backup Users

Provide a comma separated list of domain users or a domain group name that require elevated privileges to access volumes created by Cloud Volumes Service.

Accountnames
administrator,cvs-svc

Security Privilege Users

Provide a list of comma separated domain user accounts that require elevated privileges to manage security log for the Active Directory associated with Cloud Volumes Service.

Accountnames
administrator,cvs-svc

You can view Cloud Volumes Service local group memberships through the MMC with the proper privileges. The following figure shows users that have been added by using the Cloud Volumes Service console.
The following table shows the list of default BUILTIN groups and what users/groups are added by default.

<table>
<thead>
<tr>
<th>Local/BUILTIN group</th>
<th>Default members</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUILTIN\Administrators*</td>
<td>DOMAIN\Domain Admins</td>
</tr>
<tr>
<td>BUILTIN\Backup Operators*</td>
<td>None</td>
</tr>
<tr>
<td>BUILTIN\Guests</td>
<td>DOMAIN\Domain Guests</td>
</tr>
<tr>
<td>BUILTIN\Power Users</td>
<td>None</td>
</tr>
<tr>
<td>BUILTIN\Domain Users</td>
<td>DOMAIN\Domain Users</td>
</tr>
</tbody>
</table>

*Group membership controlled in Cloud Volumes Service Active Directory connection configuration.
You can view local users and groups (and group members) in the MMC window, but you cannot add or delete objects or change group memberships from this console. By default, only the Domain Admins group and Administrator are added to the BUILTIN\Administrators group in Cloud Volumes Service. Currently, you cannot modify this.
MMC/Computer Management access

SMB access in Cloud Volumes Service provides connectivity to the Computer Management MMC, which allows you to view shares, manage share ACLs, and view/manage SMB sessions and open files.

To use the MMC to view SMB shares and sessions in Cloud Volumes Service, the user logged in currently must be a domain administrator. Other users are allowed access to view or manage the SMB server from MMC and receive a You Do Not Have Permissions dialog box when attempting to view shares or sessions on the Cloud Volumes Service SMB instance.

To connect to the SMB server, open Computer Management, right click Computer Management and then select Connect To Another Computer. This opens the Select Computer dialog box where you can enter the SMB server name (found in the Cloud Volumes Service volume information).

When you view SMB shares with the proper permissions, you see all available shares in the Cloud Volumes Service instance that share the Active Directory connection. To control this behavior, set the Hide SMB Shares option on the Cloud Volumes Service volume instance.

Remember, only one Active Directory connection is allowed per region.
The following table shows a list of supported/unsupported functionality for the MMC.

<table>
<thead>
<tr>
<th>Supported functions</th>
<th>Unsupported functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>• View shares</td>
<td>• Creating new local users/groups</td>
</tr>
<tr>
<td>• View active SMB sessions</td>
<td>• Managing/viewing existing local user/groups</td>
</tr>
<tr>
<td>• View open files</td>
<td>• View events or performance logs</td>
</tr>
<tr>
<td>• View local users and groups</td>
<td>• Managing storage</td>
</tr>
<tr>
<td>• View local group memberships</td>
<td>• Managing services and applications</td>
</tr>
<tr>
<td>• Enumerate the list of sessions, files, and tree connections in the system</td>
<td></td>
</tr>
<tr>
<td>• Close open files in the system</td>
<td></td>
</tr>
<tr>
<td>• Close open sessions</td>
<td></td>
</tr>
<tr>
<td>• Create/manage shares</td>
<td></td>
</tr>
</tbody>
</table>

**SMB server security information**

The SMB server in Cloud Volumes Service uses a series of options that define security policies for SMB connections, including things such as Kerberos clock skew, ticket age, encryption, and more.

The following table contains a list of those options, what they do, the default configurations, and if they can be modified with Cloud Volumes Service. Some options do not apply to Cloud Volumes Service.
<table>
<thead>
<tr>
<th>Security option</th>
<th>What it does</th>
<th>Default value</th>
<th>Can change?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Kerberos Clock Skew (minutes)</td>
<td>Maximum time skew between Cloud Volumes Service and domain controllers. If the time skew exceeds 5 minutes, Kerberos authentication fails. This is set to the Active Directory default value.</td>
<td>5</td>
<td>No</td>
</tr>
<tr>
<td>Kerberos Ticket Lifetime (hours)</td>
<td>Maximum time a Kerberos ticket remains valid before requiring a renewal. If no renewal occurs before the 10 hours, you must obtain a new ticket. Cloud Volumes Service performs these renewals automatically. 10 hours is the Active Directory default value.</td>
<td>10</td>
<td>No</td>
</tr>
<tr>
<td>Maximum Kerberos Ticket Renewal (days)</td>
<td>Maximum number of days that a Kerberos ticket can be renewed before a new authorization request is needed. Cloud Volumes Service automatically renews tickets for SMB connections. Seven days is the Active Directory default value.</td>
<td>7</td>
<td>No</td>
</tr>
<tr>
<td>Kerberos KDC Connection Timeout (secs)</td>
<td>The number of seconds before a KDC connection times out.</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>Require Signing for Incoming SMB Traffic</td>
<td>Setting to require signing for SMB traffic. If set to true, clients that do not support signing fail connectivity.</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>Require Password Complexity for Local User Accounts</td>
<td>Used for passwords on local SMB users. Cloud Volumes Service does not support local user creation, so this option does not apply to Cloud Volumes Service.</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Security option</td>
<td>What it does</td>
<td>Default value</td>
<td>Can change?</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Use start_tls for Active Directory LDAP Connections</td>
<td>Used to enable start TLS connections for Active Directory LDAP. Cloud Volumes Service does not currently support enabling this.</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Is AES-128 and AES-256 Encryption for Kerberos Enabled</td>
<td>This controls whether AES encryption is used for Active Directory connections and is controlled with the Enable AES Encryption for Active Directory Authentication option when creating/modifying the Active Directory connection.</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>LM Compatibility Level</td>
<td>Level of supported authentication dialects for Active Directory connections. See the section “SMB authentication dialects” for more information.</td>
<td>ntlmv2-krb</td>
<td>No</td>
</tr>
<tr>
<td>Require SMB Encryption for Incoming CIFS Traffic</td>
<td>Requires SMB encryption for all shares. This is not used by Cloud Volumes Service; instead, set encryption on a per-volume basis (see the section “SMB share security features”).</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Client Session Security</td>
<td>Sets signing and/or sealing for LDAP communication. This is not currently set in Cloud Volumes Service but might be needed in future releases to address . Remediation for LDAP authentication issues due to the Windows patch is covered in the section “LDAP channel binding.”.</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>SMB2 enable for DC connections</td>
<td>Uses SMB2 for DC connections. Enabled by default.</td>
<td>System-default</td>
<td>No</td>
</tr>
<tr>
<td>Security option</td>
<td>What it does</td>
<td>Default value</td>
<td>Can change?</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LDAP Referral Chasing</td>
<td>When using multiple LDAP servers, referral chasing allows the client to refer to other LDAP servers in the list when an entry is not found in the first server. This is currently not supported by Cloud Volumes Service.</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Use LDAPS for Secure Active Directory Connections</td>
<td>Enables the use of LDAP over SSL. Currently not supported by Cloud Volumes Service.</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Encryption is required for DC Connection</td>
<td>Requires encryption for successful DC connections. Disabled by default in Cloud Volumes Service.</td>
<td>False</td>
<td>No</td>
</tr>
</tbody>
</table>

**Dual-protocol/multiprotocol**

Cloud Volumes Service offers the ability to share the same datasets to both SMB and NFS clients while maintaining proper access permissions (dual-protocol). This is done by coordinating identity mapping between protocols and using a centralized backend LDAP server to provide the UNIX identities to Cloud Volumes Service. You can use Windows Active Directory to provide both Windows and UNIX users for ease of use.

**Access control**

- **Share access controls.** Determine which clients and/or user and groups can access a NAS share. For NFS, export policies and rules control client access to exports. NFS exports are managed from the Cloud Volumes Service instance. SMB makes use of CIFS/SMB shares and share ACLs to provide more granular control at the user and group level. You can only configure share-level ACLs from SMB clients by using MMC/Computer Management with an account that has administrator rights on the Cloud Volumes Service instance (see the section “Accounts with local/BUILTIN administrator/backup rights.”).

- **File access controls.** Control permissions at a file or folder level and are always managed from the NAS client. NFS clients can make use of traditional mode bits (rwx) or NFSv4 ACLs. SMB clients leverage NTFS permissions.

The access control for volumes that serve data to both NFS and SMB depends on the protocol in use. For information on permissions with dual protocol, see the section “Permission model.”

**User mapping**

When a client accesses a volume, Cloud Volumes Service attempts to map the incoming user to a valid user in the opposite direction. This is necessary for proper access to be determined across protocols and to ensure that the user requesting access is indeed who they claim to be.

For example, if a Windows user named joe attempts access to a volume with UNIX permissions through SMB,
then Cloud Volumes Service performs a search to find a corresponding UNIX user named joe. If one exists, then files that are written to an SMB share as Windows user joe appears as UNIX user joe from NFS clients.

Alternately, if a UNIX user named joe attempts access to a Cloud Volumes Service volume with Windows permissions, then the UNIX user must be able to map to a valid Windows user. Otherwise, access to the volume is denied.

Currently, only Active Directory is supported for external UNIX identity management with LDAP. For more information about configuring access to this service, see Creating an AD connection.

**Permission model**

When using dual-protocol setups, Cloud Volumes Service makes use of security styles for volumes to determine the type of ACL. These security styles are set based on which NAS protocol is specified, or in the case of dual protocol, is a choice made at the time of Cloud Volumes Service volume creation.

- If you are only using NFS, Cloud Volumes Service volumes use UNIX permissions.
- If you are only using SMB, Cloud Volumes Service volumes use NTFS permissions.

If you are creating a dual-protocol volume, you can choose the ACL style at volume creation. This decision should be made based on the desired permissions management. If your users manage permissions from Windows/SMB clients, select NTFS. If your users prefer using NFS clients and chmod/chown, use UNIX security styles.

**Considerations for creating Active Directory connections**

Cloud Volumes Service provides the ability to connect your Cloud Volumes Service instance to an external Active Directory server for identity management for both SMB and UNIX users. Creating an Active Directory connection is required to use SMB in Cloud Volumes Service.

The configuration for this provides several options that require some consideration for security. The external Active Directory server can be an on-premises instance or cloud native. If you are using an on-premises Active Directory server, don’t expose the domain to the external network (such as with a DMZ or an external IP address). Instead, use secure private tunnels or VPNs, one-way forest trusts, or dedicated network connections to the on-premises networks with Private Google Access. See the Google Cloud documentation for more information about best practices using Active Directory in Google Cloud.

CVS-SW requires Active Directory servers to be located in the same region. If a DC connection is attempted in CVS-SW to another region, the attempt fails. When using CVS-SW, be sure to create Active Directory sites that include the Active Directory DCs and then specify sites in Cloud Volumes Service to avoid cross-region DC connection attempts.

**Active Directory credentials**

When SMB or LDAP for NFS is enabled, Cloud Volumes Service interacts with the Active Directory controllers to create a machine account object to use for authentication. This is no different from how a Windows SMB client joins a domain and requires the same access rights to Organizational Units (OUs) in Active Directory.

In many cases, security groups do not allow the use of a Windows administrator account on external servers such as Cloud Volumes Service. In some cases, the Windows Administrator user is disabled entirely as a security best practice.
Permissions needed to create SMB machine accounts

To add Cloud Volumes Service machine objects to an Active Directory, an account that either has administrative rights to the domain or has delegated permissions to create and modify machine account objects to a specified OU is required. You can do this with the Delegation of Control Wizard in Active Directory by creating a custom task that provides a user access to creation/deletion of computer objects with the following access permissions provided:

- Read/Write
- Create/Delete All Child Objects
- Read/Write All Properties
- Change/Reset Password

Doing this automatically adds a security ACL for the defined user to the OU in Active Directory and minimizes the access to the Active Directory environment. After a user has been delegated, that username and password can be provided as Active Directory Credentials in this window.

The username and password that is passed to the Active Directory domain leverages Kerberos encryption during the machine account object query and creation for added security.

Active Directory connection details

The Active Directory Connection Details provide fields for administrators to give specific Active Directory schema information for machine account placement, such as the following:

- **Active Directory Connection Type.** Used to specify whether the Active Directory connection in a region is used for volumes of either Cloud Volumes Service or CVS-Performance service type. If this is set incorrectly on an existing connection, it might not work properly when used or edited.

- **Domain.** The Active Directory domain name.

- **Site.** Limits Active Directory servers to a specific site for security and performance considerations. This is necessary when multiple Active Directory servers span regions because Cloud Volumes Service does not currently support allowing Active Directory authentication requests to Active Directory servers in a different region than the Cloud Volumes Service instance. (For instance, the Active Directory domain controller is in a region that only CVS-Performance supports but you want an SMB share in a CVS-SW instance.)

- **DNS servers.** DNS servers to use in name lookups.

- **NetBIOS name (optional).** If desired, the NetBIOS name for the server. This what is used when new machine accounts are created using the Active Directory connection. For instance, if the NetBIOS name is set to CVS-EAST then the machine account names will be CVS-EAST-{1234}. See the section "How Cloud Volumes Service shows up in Active Directory" for more information.

- **Organizational Unit (OU).** The specific OU to create the computer account. This is useful if you're delegating control to a user for machine accounts to a specific OU.

- **AES Encryption.** You can also check or uncheck the Enable AES Encryption for AD Authentication checkbox. Enabling AES encryption for Active Directory authentication provides extra security for Cloud Volumes Service to Active Directory communication during user and group lookups. Before enabling this option, check with your domain administrator to confirm that the Active Directory domain controllers support AES authentication.
By default, most Windows servers do not disable weaker ciphers (such as DES or RC4-HMAC), but if you choose to disable weaker ciphers, confirm Cloud Volumes Service Active Directory connection has been configured to enable AES. Otherwise, authentication failures occur. Enabling AES encryption doesn’t disable weaker ciphers but instead adds support for AES ciphers to the Cloud Volumes Service SMB machine account.

Kerberos realm details

This option does not apply to SMB servers. Rather, it is used when configuring NFS Kerberos for the Cloud Volumes Service system. When these details are populated, the NFS Kerberos realm is configured (similar to a krb5.conf file on Linux) and is used when NFS Kerberos is specified on the Cloud Volumes Service volume creation, as the Active Directory connection acts as the NFS Kerberos Distribution Center (KDC).

Non-Windows KDCs are currently unsupported for use with Cloud Volumes Service.

Region

A region enables you to specify the location where the Active Directory connection resides. This region must be the same region as the Cloud Volumes Service volume.

- **Local NFS Users with LDAP.** In this section, there is also an option to Allow Local NFS Users with LDAP. This option must be left unselected if you want to extend your UNIX user group membership support beyond the 16-group limitation of NFS (extended groups). However, using extended groups requires a configured LDAP server for UNIX identities. If you don’t have an LDAP server, leave this option unselected. If you have an LDAP server and want to also use local UNIX users (such as root), select this option.

Backup users

This option enables you to specify Windows users that have backup permissions to the Cloud Volumes Service volume. Backup privileges (SeBackupPrivilege) are necessary for some applications to properly backup and restore data in NAS volumes. This user has a high level of access to data in the volume, so you should consider enabling auditing of that user access. After it is enabled, audit events display in Event Viewer > Windows Logs > Security.
Security privilege users

This option enables you to specify Windows users that have security modification permissions to the Cloud Volumes Service volume. Security privileges (SeSecurityPrivilege) are necessary for some applications (such as SQL Server) to properly set permissions during installation. This privilege is needed to manage the security log. Although this privilege is not as powerful as SeBackupPrivilege, NetApp recommends auditing user access of users with this privilege level if needed.

For more information, see Special privileges assigned to new logon.

How Cloud Volumes Service shows up in Active Directory

Cloud Volumes Service shows up in Active Directory as a normal machine account object. The naming conventions are as follows.

- CIFS/SMB and NFS Kerberos create separate machine account objects.
- NFS with LDAP enabled creates a machine account in Active Directory for Kerberos LDAP binds.
- Dual protocol volumes with LDAP share the CIFS/SMB machine account for LDAP and SMB.
- CIFS/SMB machine accounts use a naming convention of NAME-1234 (random four digit ID with hyphen appended to <10 character name) for the machine account. You can define NAME by the NetBIOS name setting on the Active Directory connection (see the section “Active Directory connection details”).
- NFS Kerberos uses NFS-NAME-1234 as the naming convention (up to 15 characters). If more than 15 characters are used, the name is NFS-TRUNCATED-NAME-1234.
• NFS-only CVS-Performance instances with LDAP enabled create an SMB machine account for binding to the LDAP server with the same naming convention as CIFS/SMB instances.

• When an SMB machine account is created, default hidden admin shares (see the section “Default hidden shares”) are also created (c$, admin$, ipc$), but those shares have no ACLs assigned and are inaccessible.

• The machine account objects are placed in CN=Computers by default, but a you can specify a different OU when necessary. See the section “Permissions needed to create SMB machine accounts” for information about what access rights are needed to add/remove machine account objects for Cloud Volumes Service.

When Cloud Volumes Service adds the SMB machine account to Active Directory, the following fields are populated:

• cn (with the specified SMB server name)
• dNSHostName (with SMBserver.domain.com)
• msDS-SupportedEncryptionTypes (Allows DES_CBC_MD5, RC4_HMAC_MD5 if AES encryption is not enabled; if AES encryption is enabled, DES_CBC_MD5, RC4_HMAC_MD5, AES128_CTS_HMAC_SHA1_96, AES256_CTS_HMAC_SHA1_96 are allowed for Kerberos ticket exchange with the machine account for SMB)
• name (with the SMB server name)
• sAMAccountName (with SMBserver$)
• servicePrincipalName (with host/smbserver.domain.com and host/smbserver SPNs for Kerberos)

If you want to disable weaker Kerberos encryption types (enctype) on the machine account, you can change the msDS-SupportedEncryptionTypes value on the machine account to one of the values in the following table to allow AES only.

<table>
<thead>
<tr>
<th>msDS-SupportedEncryptionTypes value</th>
<th>Enctype enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>DES_CBC_MD5</td>
</tr>
<tr>
<td>4</td>
<td>RC4_HMAC</td>
</tr>
<tr>
<td>8</td>
<td>AES128_CTS_HMAC_SHA1_96 only</td>
</tr>
<tr>
<td>16</td>
<td>AES256_CTS_HMAC_SHA1_96 only</td>
</tr>
<tr>
<td>24</td>
<td>AES128_CTS_HMAC_SHA1_96 and AES256_CTS_HMAC_SHA1_96</td>
</tr>
<tr>
<td>30</td>
<td>DES_CBC_MD5, RC4_HMAC, AES128_CTS_HMAC_SHA1_96 and AES256_CTS_HMAC_SHA1_96</td>
</tr>
</tbody>
</table>

To enable AES encryption for SMB machine accounts, click Enable AES Encryption for AD Authentication when creating the Active Directory connection.

To enable AES encryption for NFS Kerberos, see the Cloud Volumes Service documentation.

Other NAS Infrastructure service dependencies (KDC, LDAP, and DNS)

When using Cloud Volumes Service for NAS shares, there might be external dependencies required for proper functionality. These dependencies are in play under specific circumstances. The following table shows various configuration options and what,
if any, dependencies are required.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Dependencies required</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFSv3 only</td>
<td>None</td>
</tr>
<tr>
<td>NFSv3 Kerberos only</td>
<td>Windows Active Directory: * KDC * DNS * LDAP</td>
</tr>
<tr>
<td>NFSv4.1 only</td>
<td>Client ID mapping configuration (/etc/idmap.conf)</td>
</tr>
<tr>
<td>NFSv4.1 Kerberos only</td>
<td>* Client ID mapping configuration (/etc/idmap.conf)</td>
</tr>
<tr>
<td></td>
<td>* Windows Active Directory: KDC DNS LDAP</td>
</tr>
<tr>
<td>SMB only</td>
<td>Active Directory: * KDC * DNS</td>
</tr>
<tr>
<td>Multiprotocol NAS (NFS and SMB)</td>
<td>* Client ID mapping configuration (NFSv4.1 only; /etc/idmap.conf)</td>
</tr>
<tr>
<td></td>
<td>* Windows Active Directory: KDC DNS LDAP</td>
</tr>
</tbody>
</table>

**Kerberos keytab rotation/password resets for machine account objects**

With SMB machine accounts, Cloud Volumes Service schedules periodic password resets for the SMB machine account. These password resets occur using Kerberos encryption and operate on a schedule of every fourth Sunday at a random time between 11PM and 1AM. These password resets change the Kerberos key versions, rotate the keytabs stored on the Cloud Volumes Service system, and help maintain a greater level of security for SMB servers running in Cloud Volumes Service. Machine account passwords are randomized and are not known to administrators.

For NFS Kerberos machine accounts, password resets take place only when a new keytab is created/exchanged with the KDC. Currently, this is not possible to do in Cloud Volumes Service.

**Network ports for use with LDAP and Kerberos**

When using LDAP and Kerberos, you should determine the network ports in use by these services. You can find a complete list of ports in use by Cloud Volumes Service in the Cloud Volumes Service documentation on security considerations.

**LDAP**

Cloud Volumes Service acts as an LDAP client and uses standard LDAP search queries for user and group lookups for UNIX identities. LDAP is necessary if you intend to use users and groups outside the standard default users provided by Cloud Volumes Service. LDAP is also necessary if you plan on using NFS Kerberos.
with user principals (such as user1@domain.com). Currently, only LDAP using Microsoft Active Directory is supported.

To use Active Directory as a UNIX LDAP server, you must populate the necessary UNIX attributes on users and groups you intend to use for UNIX identities. Cloud Volumes Service uses a default LDAP schema template that queries attributes based on RFC-2307-bis. As a result, the following table shows the bare minimum necessary Active Directory attributes to populate for users and groups and what each attribute is used for.

For more information about setting LDAP attributes in Active Directory, see Managing dual-protocol access.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>What it does</th>
</tr>
</thead>
<tbody>
<tr>
<td>uid*</td>
<td>Specifies the UNIX user name</td>
</tr>
<tr>
<td>uidNumber*</td>
<td>Specifies the UNIX user’s numeric ID</td>
</tr>
<tr>
<td>gidNumber*</td>
<td>Specifies the UNIX user’s primary group numeric ID</td>
</tr>
<tr>
<td>objectClass*</td>
<td>Specifies what type of object is being used; Cloud Volumes Service requires “user” to be included in the list of object classes (is included in most Active Directory deployments by default).</td>
</tr>
<tr>
<td>name</td>
<td>General information about the account (real name, phone number, and so on—also known as gecos)</td>
</tr>
<tr>
<td>unixUserPassword</td>
<td>No need to set this; not used in UNIX identity lookups for NAS authentication. Setting this puts the configured unixUserPassword value in plaintext.</td>
</tr>
<tr>
<td>unixHomeDirectory</td>
<td>Defines path to UNIX home directories when a user authenticates against LDAP from a Linux client. Set this if you want to use LDAP for UNIX home directory functionality.</td>
</tr>
<tr>
<td>loginShell</td>
<td>Defines path to the bash/profile shell for Linux clients when a user authenticates against LDAP.</td>
</tr>
</tbody>
</table>

*Denotes attribute is required for proper functionality with Cloud Volumes Service. Remaining attributes are for client-side use only.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>What it does</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn*</td>
<td>Specifies the UNIX group name. When using Active Directory for LDAP, this is set when the object is first created, but it can be changed later. This name cannot be the same as other objects. For instance, if your UNIX user named user1 belongs to a group named user1 on your Linux client, Windows doesn’t allow two objects with the same cn attribute. To work around this, rename the Windows user to a unique name (such as user1-UNIX); LDAP in Cloud Volumes Service uses the uid attribute for UNIX user names.</td>
</tr>
<tr>
<td>gidNumber*</td>
<td>Specifies the UNIX group numeric ID.</td>
</tr>
<tr>
<td>Attribute</td>
<td>What it does</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>objectClass*</td>
<td>Specifies what type of object is being used; Cloud Volumes Service requires group to be included in the list of object classes (this attribute is included in most Active Directory deployments by default).</td>
</tr>
<tr>
<td>memberUid</td>
<td>Specifies which UNIX users are members of the UNIX group. With Active Directory LDAP in Cloud Volumes Service, this field is not necessary. The Cloud Volumes Service LDAP schema uses the Member field for group memberships.</td>
</tr>
<tr>
<td>Member*</td>
<td>Required for group memberships/secondary UNIX groups. This field is populated by adding Windows users to Windows groups. However, if the Windows groups don’t have UNIX attributes populated, they are not included in the UNIX user’s group membership lists. Any groups that need to be available in NFS must populate the required UNIX group attributes listed in this table.</td>
</tr>
</tbody>
</table>

*Denotes attribute is required for proper functionality with Cloud Volumes Service. Remaining attributes are for client-side use only.

**LDAP bind information**

To query users in LDAP, Cloud Volumes Service must bind (login) to the LDAP service. This login has read-only permissions and is used to query LDAP UNIX attributes for directory lookups. Currently, LDAP binds are possible only by using an SMB machine account.

You can only enable LDAP for **CVS-Performance** instances and use it for NFSv3, NFSv4.1, or dual-protocol volumes. An Active Directory connection must be established in the same region as the Cloud Volumes Service volume for successful deployment of the LDAP-enabled volume.

When LDAP is enabled, the following occurs in specific scenarios.

- If only NFSv3 or NFSv4.1 is used for the Cloud Volumes Service project, then a new machine account is created in the Active Directory domain controller, and the LDAP client in Cloud Volumes Service binds to Active Directory by using the machine account credentials. No SMB shares are created for the NFS volume and default hidden administrative shares (see the section “Default hidden shares”) have share ACLs removed.

- If dual-protocol volumes are used for the Cloud Volumes Service project, then only the single machine account created for SMB access is used to bind the LDAP client in Cloud Volumes Service to Active Directory. No additional machine accounts are created.

- If dedicated SMB volumes are created separately (either before or after NFS volumes with LDAP are enabled), then the machine account for LDAP binds is shared with the SMB machine account.

- If NFS Kerberos is also enabled, two machine accounts are created—one for SMB shares and/or LDAP binds and one for NFS Kerberos authentication.

**LDAP queries**

Although LDAP binds are encrypted, LDAP queries are passed over the wire in plaintext by using the common LDAP port 389. This well-known port cannot currently be changed in Cloud Volumes Service. As a result,
someone with access to packet sniffing in the network can see user and group names, numeric IDs, and group memberships.

However, Google Cloud VMs cannot sniff other VM’s unicast traffic. Only VMs actively participating in LDAP traffic (that is, being able to bind) can see traffic from the LDAP server. For more information about packet sniffing in Cloud Volumes Service, see the section "Packet sniffing/trace considerations."

**LDAP client configuration defaults**

When LDAP is enabled in a Cloud Volumes Service instance, an LDAP client configuration is created with specific configuration details by default. In some cases, options either do not apply to Cloud Volumes Service (not supported) or are not configurable.

<table>
<thead>
<tr>
<th>LDAP client option</th>
<th>What it does</th>
<th>Default value</th>
<th>Can change?</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP Server List</td>
<td>Sets LDAP server names or IP addresses to use for queries. This is not used for Cloud Volumes Service. Instead, Active Directory Domain is used to define LDAP servers.</td>
<td>Not set</td>
<td>No</td>
</tr>
<tr>
<td>Active Directory Domain</td>
<td>Sets the Active Directory Domain to use for LDAP queries. Cloud Volumes Service leverages SRV records for LDAP in DNS to find LDAP servers in the domain.</td>
<td>Set to the Active Directory domain specified in the Active Directory connection.</td>
<td>No</td>
</tr>
<tr>
<td>Preferred Active Directory Servers</td>
<td>Sets the preferred Active Directory servers to use for LDAP. Not supported by Cloud Volumes Service. Instead, use Active Directory sites to control LDAP server selection.</td>
<td>Not set.</td>
<td>No</td>
</tr>
<tr>
<td>Bind using SMB Server Credentials</td>
<td>Binds to LDAP by using the SMB machine account. Currently, the only supported LDAP bind method in Cloud Volumes Service.</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Schema Template</td>
<td>The schema template used for LDAP queries.</td>
<td>MS-AD-BIS</td>
<td>No</td>
</tr>
<tr>
<td>LDAP Server Port</td>
<td>The port number used for LDAP queries. Cloud Volumes Service currently uses only the standard LDAP port 389. LDAPS/port 636 is not currently supported.</td>
<td>389</td>
<td>No</td>
</tr>
<tr>
<td>LDAP client option</td>
<td>What it does</td>
<td>Default value</td>
<td>Can change?</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Is LDAPS Enabled</td>
<td>Controls whether LDAP over Secure Sockets Layer (SSL) is used for queries and binds. Currently not supported by Cloud Volumes Service.</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Query Timeout (sec)</td>
<td>Timeout for queries. If queries take longer than the specified value, queries fail.</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>Minimum Bind Authentication Level</td>
<td>The minimum supported bind level. Because Cloud Volumes Service uses machine accounts for LDAP binds and Active Directory does not support anonymous binds by default, this option does not come into play for security.</td>
<td>Anonymous</td>
<td>No</td>
</tr>
<tr>
<td>Bind DN</td>
<td>The user/distinguished name (DN) used for binds when simple bind is used. Cloud Volumes Service uses machine accounts for LDAP binds and does not currently support simple bind authentication.</td>
<td>Not set</td>
<td>No</td>
</tr>
<tr>
<td>Base DN</td>
<td>The base DN used for LDAP searches.</td>
<td>The Windows domain use for the Active Directory connection, in DN format (that is, DC=domain, DC=local).</td>
<td>No</td>
</tr>
<tr>
<td>Base search scope</td>
<td>The search scope for base DN searches. Values can include base, onelevel, or subtree. Cloud Volumes Service only supports subtree searches.</td>
<td>Subtree</td>
<td>No</td>
</tr>
<tr>
<td>User DN</td>
<td>Defines the DN where user searches start for LDAP queries. Currently not supported for Cloud Volumes Service, so all user searches start at the base DN.</td>
<td>Not set</td>
<td>No</td>
</tr>
<tr>
<td>LDAP client option</td>
<td>What it does</td>
<td>Default value</td>
<td>Can change?</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>User search scope</td>
<td>The search scope for user DN searches. Values can include base, onelevel, or subtree. Cloud Volumes Service does not support setting the user search scope.</td>
<td>Subtree</td>
<td>No</td>
</tr>
<tr>
<td>Group DN</td>
<td>Defines the DN where group searches start for LDAP queries. Currently not supported for Cloud Volumes Service, so all group searches start at the base DN.</td>
<td>Not set</td>
<td>No</td>
</tr>
<tr>
<td>Group search scope</td>
<td>The search scope for group DN searches. Values can include base, onelevel, or subtree. Cloud Volumes Service does not support setting the group search scope.</td>
<td>Subtree</td>
<td>No</td>
</tr>
<tr>
<td>Netgroup DN</td>
<td>Defines the DN where netgroup searches start for LDAP queries. Currently not supported for Cloud Volumes Service, so all netgroup searches start at the base DN.</td>
<td>Not set</td>
<td>No</td>
</tr>
<tr>
<td>Netgroup search scope</td>
<td>The search scope for netgroup DN searches. Values can include base, onelevel, or subtree. Cloud Volumes Service does not support setting the netgroup search scope.</td>
<td>Subtree</td>
<td>No</td>
</tr>
<tr>
<td>Use start_tls over LDAP</td>
<td>Leverages Start TLS for certificate based LDAP connections over port 389. Currently not supported by Cloud Volumes Service.</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Enable netgroup-by-host lookup</td>
<td>Enables netgroup lookups by hostname rather than expanding netgroups to list all members. Currently not supported by Cloud Volumes Service.</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>LDAP client option</td>
<td>What it does</td>
<td>Default value</td>
<td>Can change?</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Netgroup-by-host DN</td>
<td>Defines the DN where netgroup-by-host searches start for LDAP queries. Netgroup-by-host is currently not supported for Cloud Volumes Service.</td>
<td>Not set</td>
<td>No</td>
</tr>
<tr>
<td>Netgroup-by-host search scope</td>
<td>The search scope for netgroup-by-host DN searches. Values can include base, onelevel or subtree. Netgroup-by-host is currently not supported for Cloud Volumes Service.</td>
<td>Subtree</td>
<td>No</td>
</tr>
<tr>
<td>Client session security</td>
<td>Defines what level of session security is used by LDAP (sign, seal, or none). LDAP signing is supported by CVS-Performance, if requested by Active Directory. CVS-SW does not support LDAP signing. For both service types, sealing is currently not supported.</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>LDAP referral chasing</td>
<td>When using multiple LDAP servers, referral chasing allows the client to refer to other LDAP servers in the list when an entry is not found in the first server. This is currently not supported by Cloud Volumes Service.</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Group membership filter</td>
<td>Provides a custom LDAP search filter to be used when looking up group membership from an LDAP server. Not currently supported with Cloud Volumes Service.</td>
<td>Not set</td>
<td>No</td>
</tr>
</tbody>
</table>

Using LDAP for asymmetric name mapping

Cloud Volumes Service, by default, maps Windows users and UNIX users with identical usernames bidirectionally without special configuration. As long as Cloud Volumes Service can find a valid UNIX user (with LDAP), then 1:1 name mapping occurs. For instance, if Windows user johnsmith is used, then, if Cloud Volumes Service can find a UNIX user named johnsmith in LDAP, name mapping succeeds for that user, all files/folders created by johnsmith show the correct user ownership, and all ACLs affecting johnsmith are...
honored regardless of the NAS protocol in use. This is known as symmetric name mapping.

Asymmetric name mapping is when the Windows user and UNIX user identity don’t match. For instance, if Windows user *johnsmith* has a UNIX identity of *jsmith*, Cloud Volumes Service needs a way to be told about the variation. Because Cloud Volumes Service currently doesn’t support creation of static name mapping rules, LDAP must be used to look up the identity of the users for both Windows and UNIX identities to ensure proper ownership of files and folders and expected permissions.

By default, Cloud Volumes Service includes LDAP in the ns-switch of the instance for the name map database, so that to provide name mapping functionality by using LDAP for asymmetric names, you only need to modify some of the user/group attributes to reflect what Cloud Volumes Service looks for.

The following table shows what attributes must be populated in LDAP for asymmetric name mapping functionality. In most cases, Active Directory is already configured to do this.

<table>
<thead>
<tr>
<th>Cloud Volumes Service attribute</th>
<th>What it does</th>
<th>Value used by Cloud Volumes Service for name mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows to UNIX objectClass</td>
<td>Specifies the type of object being used. (That is, user, group, posixAccount, and so on)</td>
<td>Must include user (can contain multiple other values, if desired.)</td>
</tr>
<tr>
<td>Windows to UNIX attribute</td>
<td>that defines the Windows username at creation. Cloud Volumes Service uses this for Windows to UNIX lookups.</td>
<td>No change needed here; sAMAccountName is the same as the Windows login name.</td>
</tr>
<tr>
<td>UID</td>
<td>Defines the UNIX username.</td>
<td>Desired UNIX username.</td>
</tr>
</tbody>
</table>

Cloud Volumes Service currently does not use domain prefixes in LDAP lookups, so multiple domain LDAP environments do not function properly with LDAP namemap lookups.

The following example shows a user with the Windows name *asymmetric*, the UNIX name *unix-user*, and the behavior it follows when writing files from both SMB and NFS.

The following figure shows how LDAP attributes look from the Windows server.
From an NFS client, you can query the UNIX name but not the Windows name:

```
# id unix-user
uid=1207(unix-user) gid=1220(sharedgroup) groups=1220(sharedgroup)
# id asymmetric
id: asymmetric: no such user
```

When a file is written from NFS as `unix-user`, the following is the result from the NFS client:

```
sh-4.2$ pwd
/mnt/home/ntfssh-4.2$
touch unix-user-file
sh-4.2$ ls -la | grep unix-user
-rwx------ 1 unix-user sharedgroup 0 Feb 28 12:37 unix-user-nfs
sh-4.2$ id
uid=1207(unix-user) gid=1220(sharedgroup) groups=1220(sharedgroup)
```
From a Windows client, you can see that the owner of the file is set to the proper Windows user:

```
PS C:\> Get-Acl \demo\home\ntfs\unix-user-nfs | select Owner
Owner
-----
NTAP\asymmetric
```

Conversely, files created by the Windows user `asymmetric` from an SMB client show the proper UNIX owner, as shown in the following text.

**SMB:**

```
PS Z:\ntfs> echo TEXT > asymmetric-user-smb.txt
```

**NFS:**

```
sh-4.2$ ls -la | grep asymmetric-user-smb.txt
-rwx------ 1 unix-user sharedgroup 14 Feb 28 12:43 asymmetric-user-smb.txt
sh-4.2$ cat asymmetric-user-smb.txt
TEXT
```

**LDAP channel binding**

Because of a vulnerability with Windows Active Directory domain controllers, [Microsoft Security Advisory ADV190023](https://www.microsoft.com/security/advisory/adv190023) changes how DCs allow LDAP binds.

The impact for Cloud Volumes Service is the same as for any LDAP client. Cloud Volumes Service does not currently support channel binding. Because Cloud Volumes Service supports LDAP signing by default through negotiation, LDAP channel binding should not be an issue. If you do have issues binding to LDAP with channel binding enabled, follow the remediation steps in ADV190023 to allow LDAP binds from Cloud Volumes Service to succeed.

**DNS**

Active Directory and Kerberos both have dependencies on DNS for host name to IP/IP to host name resolution. DNS requires port 53 to be open. Cloud Volumes Service does not make any modifications to DNS records, nor does it currently support the use of dynamic DNS on network interfaces.

You can configure Active Directory DNS to restrict which servers can update DNS records. For more information, see [Secure Windows DNS](https://docs.microsoft.com/en-us/windows/security/threat-protection/dns/dns-security).

Note that resources within a Google project default to using Google Cloud DNS, which isn’t connected with Active Directory DNS. Clients using Cloud DNS cannot resolve UNC paths returned by Cloud Volumes Service. Windows clients joined to the Active Directory domain are configured to use Active Directory DNS and can resolve such UNC paths.

To join a client to Active Directory, you must configure its DNS configuration to use Active Directory DNS.
Optionally, you can configure Cloud DNS to forward requests to Active Directory DNS. See Why can’t my client resolve the SMB NetBIOS name? for more information.

Cloud Volumes Service does not currently support DNSSEC and DNS queries are performed in plaintext.

File access auditing

Currently not supported for Cloud Volumes Service.

Antivirus protection

You must perform antivirus scanning in Cloud Volumes Service at the client to a NAS share. There is currently no native antivirus integration with Cloud Volumes Service.

Service operation

The Cloud Volumes Service team manages the backend services in Google Cloud and uses multiple strategies to secure the platform and prevent unwanted access.

Each customer gets their own unique subnet that has access fenced off from other customers by default, and every tenant in Cloud Volumes Service gets their own namespace and VLAN for total data isolation. After a user is authenticated, the Service Delivery Engine (SDE) can only read configuration data specific to that tenant.

Physical security

With proper preapproval, only onsite engineers and NetApp-badged Field Support Engineers (FSEs) have access to the cage and racks for physical work. Storage and network management is not permitted. Only these onsite resources are able to perform hardware maintenance tasks.

For onsite engineers, a ticket is raised for the statement of work (SOW) that includes the rack ID and device location (RU) and all other details are included in the ticket. For NetApp FSEs, a site visitation ticket must be raised with the COLO and the ticket includes the visitor’s details, date, and time for auditing purposes. The SOW for the FSE is communicated internally to NetApp.

Operations team

The operations team for Cloud Volumes Service consists of Production Engineering and a Site Reliability Engineer (SRE) for Cloud Volume Services and NetApp Field Support Engineers and Partners for hardware. All operations team members are accredited for work in Google Cloud and detailed records of work are maintained for every ticket raised. In addition, there is a stringent change control and approval process in place to ensure each decision is appropriately scrutinized.

The SRE team manages the control plane and how the data is routed from UI requests to backend hardware and software in Cloud Volumes Service. The SRE team also manages system resources, such as volume and inode maximums. SREs are not allowed to interact with or have access to customer data. SREs also provide coordination with Return Material Authorizations (RMAs), such as new disk or memory replacement requests for the backend hardware.

Customer responsibilities

Customers of Cloud Volumes Service manage their organization’s Active Directory and user role management as well as the volume and data operations. Customers can have administrative roles and can delegate
permissions to other end users within the same Google Cloud project using the two predefined roles that NetApp and Google Cloud provide (Administrator and Viewer).

The administrator can peer any VPC within the customer project to Cloud Volumes Service that the customer determines to be appropriate. It is the responsibility of the customer to manage access to their Google Cloud marketplace subscription and to manage the VPCs that have access to the data plane.

**Malicious SRE protection**

One concern that could arise is how does Cloud Volumes Service protect against scenarios in which there is a malicious SRE or when SRE credentials have been compromised?

Access to the production environment is with a limited number of SRE individuals only. Administrative privileges are further restricted to a handful of experienced administrators. All actions performed by anyone in the Cloud Volumes Service production environment are logged and any anomalies to the baseline or suspicious activities are detected by our security information and event management (SIEM) threat intelligence platform. As a result, malicious actions can be tracked and mitigated before too much damage is done to the Cloud Volumes Service backend.

**Volume life cycle**

Cloud Volumes Service manages only the objects within the service—not the data within the volumes. Only clients accessing the volumes can manage the data, the ACLs, file owners, and so on. The data in these volumes is encrypted at rest and access is limited to tenants of the Cloud Volumes Service instance.

The volume lifecycle for Cloud Volumes Service is create-update-delete. Volumes retain Snapshot copies of volumes until the volumes are deleted, and only validated Cloud Volumes Service administrators can delete volumes in Cloud Volumes Service. When a volume deletion is requested by an administrator, an additional step of entering the volume name is required to verify the deletion. After a volume is deleted, the volume is gone and cannot be recovered.

In cases where a Cloud Volumes Service contract is terminated, NetApp marks volumes for deletion after a specific time period. Before that time period expires, you can recover volumes at the customer’s request.

**Certifications**

Cloud Volumes Services for Google Cloud is currently certified to ISO/IEC 27001:2013 and ISO/IEC 27018:2019 standards. The service also recently received its SOC2 Type I attestation report. For information about the NetApp commitment to data security and privacy, see [Compliance: Data security and data privacy](https://cloud.google.com/architecture/partners/netapp-cloud-volumes/).

**GDPR**

Our commitments to privacy and compliance with GDPR are available in a number of our customer contracts, such as our [Customer Data Processing Addendum](https://cloud.google.com/architecture/partners/netapp-cloud-volumes/), which includes the [Standard Contractual Clauses](https://cloud.google.com/architecture/partners/netapp-cloud-volumes/) provided by the European Commission. We also make these commitments in our Privacy Policy, backed by the core values set out in our corporate Code of Conduct.

**Additional information and contact information**

To learn more about the information that is described in this document, review the following documents and/or websites:

- Google Cloud documentation for Cloud Volumes Service

  [https://cloud.google.com/architecture/partners/netapp-cloud-volumes/](https://cloud.google.com/architecture/partners/netapp-cloud-volumes/)
BlueXP Backup and Recovery

BlueXP backup and recovery for VMs

3-2-1 Data Protection for VMware with SnapCenter Plug-in and BlueXP backup and recovery for VMs

Author: Josh Powell - NetApp Solutions Engineering

Overview

The 3-2-1 backup strategy is an industry accepted data protection method, providing a comprehensive approach to safeguarding valuable data. This strategy is reliable and ensures that even if some unexpected disaster strikes, there will still be a copy of the data available.

The strategy is comprised of three fundamental rules:

1. Keep at least three copies of your data. This ensures that even if one copy is lost or corrupted, you still have at least two remaining copies to fall back on.

2. Store two backup copies on different storage media or devices. Diversifying storage media helps protect against device-specific or media-specific failures. If one device gets damaged or one type of media fails, the other backup copy remains unaffected.

3. Finally, ensure that at least one backup copy is offsite. Offsite storage serves as a fail-safe against localized disasters like fires or floods that could render onsite copies unusable.

This solution document covers a 3-2-1 backups solution using SnapCenter Plug-in for VMware vSphere (SCV) to create primary and secondary backups of our on-premises virtual machines and BlueXP backup and recovery for virtual machines to backup a copy of our data to cloud storage or StorageGRID.
Use Cases

This solution addresses the following use cases:

- Backup and restore of on-premises virtual machines and datastores using using SnapCenter Plug-in for VMware vSphere.
- Backup and restore of on-premises virtual machines and datastores, hosted on ONTAP clusters, and backed up to object storage using BlueXP backup and recovery for virtual machines.

NetApp ONTAP Data Storage

ONTAP is NetApp’s industry leading storage solution that offers unified storage whether you access over SAN or NAS protocols. The 3-2-1 backup strategy ensures on-premises data is protected on more than one media type and NetApp offers platforms ranging from high-speed flash to lower-cost media.

For more information on all of NetApp’s hardware platform’s check out NetApp Data Storage.

SnapCenter Plug-in for VMware vSphere

The SnapCenter Plugin for VMware vSphere is a data protection offering which is tightly integrated with VMware vSphere and allows easy management of backup and restores for virtual machines. As part of that solution, SnapMirror provides a fast and reliable method to create a second immutable backup copy of virtual machine data on a secondary ONTAP storage cluster. With this architecture in place, virtual machine restore operations can easily be initiated from either the primary or secondary backup locations.

SCV is deployed as a linux virtual appliance using an OVA file. The plug-in now uses a remote plug-in architecture. The remote plug-in runs outside of the vCenter server and is hosted on the SCV virtual appliance.

For detailed information on SCV refer to SnapCenter Plug-in for VMware vSphere documentation.

BlueXP backup and recovery for virtual machines

BlueXP backup and recovery is a cloud based tool for data management that provides a single control plane for a wide range of backup and recovery operations across both on-premises and cloud environments. Part of the NetApp BlueXP backup and recovery suite is a feature that integrates with the SnapCenter Plugin for VMware vSphere (on-premises) to extend a copy of the data to object storage in the cloud. This establishes a third copy of the data offsite that is sourced from the primary or secondary storage backups. BlueXP backup and recovery makes it easy to set up storage policies that transfer copies of your data from either of these two on-prem locations.

Choosing between the primary and secondary backups as the source in BlueXP Backup and Recovery will result in one of two topologies being implemented:
Fan-out Topology – When a backup is initiated by the SnapCenter Plug-in for VMware vSphere, a local snapshot is immediately taken. SCV then initiates a SnapMirror operation that replicates the most recent snapshot to the Secondary ONTAP cluster. In BlueXP Backup and Recovery, a policy specifies the primary ONTAP cluster as the source for a snapshot copy of the data to be transferred to object storage in your cloud provider of choice.

![Fan-out Topology Diagram](image)

Cascading Topology – Creating the primary and secondary data copies using SCV is identical to the fan-out topology mentioned above. However, this time a policy is created in BlueXP Backup and Recovery specifying that the backup to object storage will originate from the secondary ONTAP cluster.

![Cascading Topology Diagram](image)

BlueXP backup and recovery can create backup copies of on-premises ONTAP snapshots to AWS Glacier, Azure Blob, and GCP Archive storage.
In addition, you can use NetApp StorageGRID as the object storage backup target. For more on StorageGRID refer to the StorageGRID landing page.

Solution Deployment Overview

This list provides the high level steps necessary to configure this solution and execute backup and restore operations from SCV and BlueXP backup and recovery:

1. Configure SnapMirror relationship between the ONTAP clusters to be used for primary and secondary data copies.
2. Configure SnapCenter Plug-In for VMware vSphere.
   a. Add Storage Systems
   b. Create backup policies
   c. Create resource groups
   d. Run backup first backup jobs
3. Configure BlueXP backup and recovery for virtual machines
   a. Add working environment
   b. Discover SCV and vCenter appliances
   c. Create backup policies
   d. Activate backups
4. Restore virtual machines from primary and secondary storage using SCV.
5. Restore virtual machines from object storage using BlueXP backup and restore.
Prerequisites

The purpose of this solution is to demonstrate data protection of virtual machines running in VMware vSphere and located on NFS Datastores hosted by NetApp ONTAP. This solution assumes the following components are configured and ready for use:

1. ONTAP storage cluster with NFS or VMFS datastores connected to VMware vSphere. Both NFS and VMFS datastores are supported. NFS datastores were utilized for this solution.
2. Secondary ONTAP storage cluster with SnapMirror relationships established for volumes used for NFS datastores.
3. BlueXP connector installed for cloud provider used for object storage backups.
4. Virtual machines to be backed are on NFS datastores residing on the primary ONTAP storage cluster.
5. Network connectivity between the BlueXP connector and on-premises ONTAP storage cluster management interfaces.
6. Network connectivity between the BlueXP connector and on-premises SCV appliance VM and between the BlueXP connector and vCenter.
7. Network connectivity between the on-premises ONTAP intercluster LIFs and the object storage service.
8. DNS configured for management SVM on primary and secondary ONTAP storage clusters. For more information refer to Configure DNS for host-name resolution.

High Level Architecture

The testing / validation of this solution was performed in a lab that may or may not match the final deployment environment.
Solution Deployment

In this solution, we provide detailed instructions for deploying and validating a solution that utilizes SnapCenter Plug-in for VMware vSphere, along with BlueXP backup and recovery, to perform the backup and recovery of Windows and Linux virtual machines within a VMware vSphere cluster located in an on-premises data center. The virtual machines in this setup are stored on NFS datastores hosted by an ONTAP A300 storage cluster. Additionally, a separate ONTAP A300 storage cluster serves as a secondary destination for volumes replicated using SnapMirror. Furthermore, object storage hosted on Amazon Web Services and Azure Blob were employed as targets for a third copy of the data.

We will go over creating SnapMirror relationships for secondary copies of our backups managed by SCV and configuration of backup jobs in both SCV and BlueXP backup and recovery.

For detailed information on SnapCenter Plug-in for VMware vSphere refer to the SnapCenter Plug-in for VMware vSphere documentation.

For detailed information on BlueXP backup and recovery refer to the BlueXP backup and recovery documentation.

Establish SnapMirror relationships between ONTAP Clusters

SnapCenter Plug-in for VMware vSphere uses ONTAP SnapMirror technology to manage the transport of secondary SnapMirror and/or SnapVault copies to a secondary ONTAP Cluster.

SCV backup policies have the option of using SnapMirror or SnapVault relationships. The primary difference is that when using the SnapMirror option, the retention schedule configured for backups in the policy will be the same at the primary and secondary locations. SnapVault is designed for archiving and when using this option a separate retention schedule can be established with the SnapMirror relationship for the snapshot copies on the secondary ONTAP storage cluster.

Setting up SnapMirror relationships can be done in BlueXP where many of the steps are automated, or it can be done using System Manager and the ONTAP CLI. All of these methods are discussed below.

Establish SnapMirror relationships with BlueXP

The following steps must be completed from the BlueXP web console:
Replication setup for primary and secondary ONTAP storage systems

Begin by logging into the BlueXP web console and navigating to the Canvas.

1. Drag and drop the source (primary) ONTAP storage system onto the destination (secondary) ONTAP storage system.

2. From the menu that appears select Replication.

3. On the Destination Peering Setup page select the destination Intercluster LIFs to be used for the connection between storage systems.
4. On the **Destination Volume Name** page, first select the source volume and then fill out the destination volume name and select the destination SVM and aggregate. Click on **Next** to continue.
5. Choose the max transfer rate for replication to occur at.

**Max Transfer Rate**

You should limit the transfer rate. An unlimited rate might negatively impact the performance of other applications and it might impact your internet performance.

- [ ] Limited to: **100** MB/s
- [ ] Unlimited (recommended for DR only machines)

6. Choose the policy that will determine the retention schedule for secondary backups. This policy can be created beforehand (see the manual process below in the **Create a snapshot retention policy** step) or can be changed after the fact if desired.
7. Finally, review all information and click on the Go button to start the replication setup process.

Establish SnapMirror relationships with System Manager and ONTAP CLI

All required steps for establishing SnapMirror relationships can be accomplished with System Manager or the ONTAP CLI. The following section provides detailed information for both methods:

Record the source and destination Intercluster logical interfaces

For the source and destination ONTAP clusters, you can retrieve the inter-cluster LIF information from System Manager or from the CLI.

1. In ONTAP System Manager, navigate to the Network Overview page and retrieve the IP addresses of Type: Intercluster that are configured to communicate with the AWS VPC where FSx is installed.

2. To retrieve the Intercluster IP addresses using the CLI run the following command:

   ONTAP-Dest::> network interface show -role intercluster
Establish cluster peering between ONTAP clusters

To establish cluster peering between ONTAP clusters, a unique passphrase entered at the initiating ONTAP cluster must be confirmed in the other peer cluster.

1. Set up peering on the destination ONTAP cluster using the `cluster peer create` command. When prompted, enter a unique passphrase that is used later on the source cluster to finalize the creation process.

   ```
   ONTAP-Dest::> cluster peer create -address-family ipv4 -peer-addrs source_intercluster_1, source_intercluster_2
   Enter the passphrase:
   Confirm the passphrase:
   ```

2. At the source cluster, you can establish the cluster peer relationship using either ONTAP System Manager or the CLI. From ONTAP System Manager, navigate to Protection > Overview and select Peer Cluster.
3. In the Peer Cluster dialog box, fill out the required information:
   a. Enter the passphrase that was used to establish the peer cluster relationship on the destination ONTAP cluster.
b. Select **Yes** to establish an encrypted relationship.

c. Enter the intercluster LIF IP address(es) of the destination ONTAP cluster.

d. Click **Initiate Cluster Peering** to finalize the process.

4. Verify the status of the cluster peer relationship from the destination ONTAP cluster with the following command:

```bash
ONTAP-Dest::> cluster peer show
```
Establish SVM peering relationship

The next step is to set up an SVM relationship between the destination and source storage virtual machines that contain the volumes that will be in SnapMirror relationships.

1. From the source FSx cluster, use the following command from the CLI to create the SVM peer relationship:

   ONTAP-Dest::> vserver peer create -vserver DestSVM -peer-vserver Backup -peer-cluster OnPremSourceSVM -applications snapmirror

2. From the source ONTAP cluster, accept the peering relationship with either ONTAP System Manager or the CLI.

3. From ONTAP System Manager, go to Protection > Overview and select Peer Storage VMs under Storage VM Peers.

4. In the Peer Storage VM's dialog box, fill out the required fields:
   - The source storage VM
   - The destination cluster
   - The destination storage VM
5. Click Peer Storage VMs to complete the SVM peering process.
Create a snapshot retention policy

SnapCenter manages retention schedules for backups that exist as snapshot copies on the primary storage system. This is established when creating a policy in SnapCenter. SnapCenter does not manage retention policies for backups that are retained on secondary storage systems. These policies are managed separately through a SnapMirror policy created on the secondary FSx cluster and associated with the destination volumes that are in a SnapMirror relationship with the source volume.

When creating a SnapCenter policy, you have the option to specify a secondary policy label that is added to the SnapMirror label of each snapshot generated when a SnapCenter backup is taken.

On the secondary storage, these labels are matched to policy rules associated with the destination volume for the purpose of enforcing retention of snapshots.

The following example shows a SnapMirror label that is present on all snapshots generated as part of a policy used for daily backups of our SQL Server database and log volumes.

For more information on creating SnapCenter policies for a SQL Server database, see the SnapCenter documentation.

You must first create a SnapMirror policy with rules that dictate the number of snapshot copies to retain.

1. Create the SnapMirror Policy on the FSx cluster.

   ONTAP-Dest::> snapmirror policy create -vserver DestSVM -policy PolicyName -type mirror-vault -restart always

2. Add rules to the policy with SnapMirror labels that match the secondary policy labels specified in the SnapCenter policies.

   ONTAP-Dest::> snapmirror policy add-rule -vserver DestSVM -policy PolicyName -snapmirror-label SnapMirrorLabelName -keep #ofSnapshotsToRetain

The following script provides an example of a rule that could be added to a policy:
Create destination volumes

To create a destination volume on ONTAP that will be the recipient of snapshot copies from our source volumes, run the following command on the destination ONTAP cluster:

```
ONTAP-Dest::> volume create -vserver DestSVM -volume DestVolName -aggregate DestAggrName -size VolSize -type DP
```

Create the SnapMirror relationships between source and destination volumes

To create a SnapMirror relationship between a source and destination volume, run the following command on the destination ONTAP cluster:

```
ONTAP-Dest::> snapmirror create -source-path OnPremSourceSVM:OnPremSourceVol -destination-path DestSVM:DestVol -type XDP -policy PolicyName
```

Initialize the SnapMirror relationships

Initialize the SnapMirror relationship. This process initiates a new snapshot generated from the source volume and copies it to the destination volume.

To create a volume, run the following command on the destination ONTAP cluster:

```
ONTAP-Dest::> snapmirror initialize -destination-path DestSVM:DestVol
```

Configure the SnapCenter Plug-in for VMware vSphere

Once installed, the SnapCenter Plug-in for VMware vSphere can be accessed from the vCenter Server Appliance Management interface. SCV will manage backups for the NFS datastores mounted to the ESXi hosts and that contain the Windows and Linux VMs.

Review the Data protection workflow section of the SCV documentation for more information on the steps involved in configuring backups.
To configure backups of your virtual machines and datastores the following steps will need to be completed from the plug-in interface.
Discovery ONTAP storage systems

Discover the ONTAP storage clusters to be used for both primary and secondary backups.

1. In the SnapCenter Plug-in for VMware vSphere navigate to **Storage Systems** in the left-hand menu and click on the **Add** button.

2. Fill out the credentials and platform type for the primary ONTAP storage system and click on **Add**.
3. Repeat this procedure for the secondary ONTAP storage system.
Create SCV backup policies

Policies specify the retention period, frequency and replication options for the backups managed by SCV.

Review the Create backup policies for VMs and datastores section of the documentation for more information.

To create backup policies complete the following steps:

1. In the SnapCenter Plug-in for VMware vSphere navigate to Policies in the left-hand menu and click on the Create button.

2. Specify a name for the policy, retention period, frequency and replication options, and snapshot label.
When creating a policy in the SnapCenter Plug-in you will see options for SnapMirror and SnapVault. If you choose SnapMirror, the retention schedule specified in the policy will be the same for both the primary and secondary snapshots. If you choose SnapVault, the retention schedule for the secondary snapshot will be based on a separate schedule implemented with the SnapMirror relationship. This is useful when you wish longer retention periods for secondary backups.

Snapshot labels are useful in that they can be used to enact policies with a specific retention period for the SnapVault copies replicated to the secondary ONTAP cluster. When SCV is used with BlueXP Backup and Restore, the Snapshot label field must either be blank or match the label specified in the BlueXP backup policy.

3. Repeat the procedure for each policy required. For example, separate policies for daily, weekly, and monthly backups.
Create resource groups

Resource groups contain the datastores and virtual machines to be included in a backup job, along with the associated policy and backup schedule.

Review the Create resource groups section of the documentation for more information.

To create resource groups complete the following steps.

1. In the SnapCenter Plug-in for VMware vSphere navigate to Resource Groups in the left-hand menu and click on the Create button.

2. In the Create Resource Group wizard, enter a name and description for the group, as well as information required to receive notifications. Click on Next.

3. On the next page select the datastores and virtual machines that wish to be included in the backup job and then click on Next.

SnapCenter Plug-in for VMware vSphere

Dashboard
Settings
Resource Groups
Policies
Storage Systems
Guest File Restore

Resource Groups

Name
SMBC
Oracle_Servers
Demo
SQL_Servers_Weekly

Available entities

Demo
DemoDS
destination
esxi7-hc-01 Local
esxi7-hc-02 Local
esxi7-hc-03 Local

Selected entities

NFS_SCV
NFS_WKLD
You have the option to select specific VMs or entire datastores. Regardless of which you choose, the entire volume (and datastore) is backed up since the backup is the result of taking a snapshot of the underlying volume. In most cases, it is easiest to choose the entire datastore. However, if you wish to limit the list of available VMs when restoring, you can choose only a subset of VMs for backup.

4. Choose options for spanning datastores for VMs with VMDKs that reside on multiple datastores and then click on **Next**.

### Create Resource Group

- **1. General info & notification**
- **2. Resource**
- **3. Spanning disks**
  - **4. Policies**
  - **5. Schedules**
  - **6. Summary**

- **Always exclude all spanning datastores**
  - This means that only the datastores directly added to the resource group and the primary datastore of VMs directly added to the resource group will be backed up.

- **Always include all spanning datastores**
  - All datastores spanned by all included VMs are included in this backup.

- **Manually select the spanning datastores to be included**
  - You will need to modify the list every time new VMs are added.

There are no spanned entities in the selected virtual entities list.

- **BlueXP backup and recovery does not currently support backing up VMs with VMDKs that span multiple datastores.**

5. On the next page select the policies that will be associated with the resource group and click on **Next**.

### Create Resource Group

- **1. General info & notification**
- **2. Resource**
- **3. Spanning disks**
- **4. Policies**
- **5. Schedules**
- **6. Summary**

When backing up SCV managed snapshots to object storage using BlueXP backup and recovery, each resource group can only be associated with a single policy.

6. Select a schedule that will determine at what times the backups will run. Click on **Next**.
7. Finally, review the summary page and then on Finish to complete the resource group creation.
Run a backup job

In this final step, run a backup job and monitor its progress. At least one backup job must be successfully completed in SCV before resources can be discovered from BlueXP backup and recovery.

1. In the SnapCenter Plug-in for VMware vSphere navigate to Resource Groups in the left-hand menu.
2. To initiate a backup job, select the desired resource group and click the Run Now button.

3. To monitor the backup job, navigate to Dashboard on the left hand menu. Under Recent Job Activities click on the Job ID number to monitor the job progress.
Configure Backups to Object Storage in BlueXP backup and recovery

For BlueXP to manage the data infrastructure effectively, it requires the prior installation of a Connector. The Connector executes the actions involved in discovering resources and managing data operations.

For more information on the BlueXP Connector refer to Learn about Connectors in the BlueXP documentation.

Once the connector is installed for the cloud provider being utilized, a graphic representation of the object storage will be viewable from the Canvas.

To configure BlueXP backup and recovery to backup data managed by SCV on-premises, complete the following steps:
Add working environments to the Canvas

The first step is to add the on-premises ONTAP storage systems to BlueXP

1. From the Canvas select Add Working Environment to begin.

2. Select On-Premises from the choice of locations and then click on the Discover button.

3. Fill out the credentials for the ONTAP storage system and click the Discover button to add the working environment.
Discover on-premises SCV appliance and vCenter

To discover the on-premises datastore and virtual machine resources, add info for the SCV data broker and credentials for the vCenter management appliance.

1. From the BlueXP left-hand menu selection Protection > Backup and recovery > Virtual Machines

2. From the Virtual Machines main screen access the Settings drop down menu and select SnapCenter Plug-in for VMware vSphere.
3. Click on the **Register** button and then enter the IP address and port number for the SnapCenter Plug-in appliance and the username and password for the vCenter management appliance. Click on the **Register** button to begin the discovery process.

4. The progress of jobs can be monitored from the Job Monitoring tab.

5. Once discovery is complete you will be able to view the datastores and virtual machines across all discovered SCV appliances.
Create BlueXP backup policies

In BlueXP backup and recovery for virtual machines, create policies to specify the retention period, backup source and the archival policy.

For more information on creating policies refer to Create a policy to back up datastores.

1. From the BlueXP backup and recovery for virtual machines main page, access the Settings drop down menu and select Policies.

2. Click on Create Policy to access the Create Policy for Hybrid Backup window.
   a. Add a name for the policy
   b. Select the desired retention period
   c. Select if backups will be sourced from the primary or secondary on-premises ONTAP storage system
   d. Optionally, specify after what period of time backups will be tiered to archival storage for additional cost savings.
The SnapMirror Label entered here is used to identify which backups to apply the policy too. The label name must match the label name in the corresponding on-premises SCV policy.

3. Click on Create to complete the policy creation.
Backup datastores to Amazon Web Services

The final step is to activate data protection for the individual datastores and virtual machines. The following steps outline how to activate backups to AWS.

For more information refer to Back up datastores to Amazon Web Services.

1. From the BlueXP backup and recovery for virtual machines main page, access the settings drop down for the datastore to be backed up and select **Activate Backup**.

![Datastores Table]

2. Assign the policy to be used for the data protection operation and click on **Next**.

![Assign Policy]

3. At the **Add Working Environments** page the datastore and working environment with a check mark should appear if the working environment has been previously discovered. If the working environment has not been previously discovered you can add it here. Click on **Next** to continue.

![Add Working Environments]

---

*Note: Images are placeholders and actual user interface may vary.*
4. At the **Select Provider** page click on AWS and then click on the **Next** button to continue.

5. Fill out the provider specific credential information for AWS including the AWS access key and secret key, region, and archival tier to be used. Also, select the ONTAP IP space for the on-premises ONTAP storage system. Click on **Next**.

6. Finally, review the backup job details and click on the **Activate Backup** button to initiate data protection of the datastore.
At this point data transfer may not immediately begin. BlueXP backup and recovery scans for any outstanding snapshots every hour and then transfers them to object storage.

Restoring Virtual Machines in the case of data loss

Ensuring the safeguarding of your data is only one aspect of comprehensive data protection. Equally crucial is the ability to promptly restore data from any location in the event of data loss or a ransomware attack. This capability is vital for maintaining seamless business operations and meeting recovery point objectives.

NetApp offers a highly adaptable 3-2-1 strategy, providing customized control over retention schedules at the
primary, secondary, and object storage locations. This strategy provides the flexibility to tailor data protection approaches to specific needs.

This section provides an overview of the data restoration process from both the SnapCenter Plug-in for VMware vSphere and BlueXP backup and recovery for virtual machines.

**Restoring Virtual Machines from SnapCenter Plug-in for VMware vSphere**

For this solution virtual machines were restored to original and alternate locations. Not all aspects of SCV’s data restoration capabilities will be covered in this solution. For in depth information on all that SCV has to offer refer to the [Restore VMs from backups](#) in the product documentation.
Complete the following steps to restore a virtual machine restore from primary or secondary storage.

1. From the vCenter client navigate to **Inventory > Storage** and click on the datastore that contains the virtual machines you wish to restore.

2. From the **Configure** tab click on **Backups** to access the list of available backups.

3. Click on a backup to access the list of VMs and then select a VM to restore. Click on **Restore**.

4. From the Restore wizard select to restore the entire virtual machine or a specific VMDK. Select to install to the original location or alternate location, provide VM name after restore, and destination datastore. Click **Next**.
5. Choose to backup from the primary or secondary storage location.

6. Finally, review a summary of the backup job and click on Finish to begin the restore process.

**Restoring Virtual Machines from BlueXP backup and recovery for virtual machines**

BlueXP backup and recovery for virtual machines allows restores of virtual machines to their original location. Restore functions are accessed through the BlueXP web console.

For more information refer to [Restore virtual machines data from the cloud](#).
To restore a virtual machine from BlueXP backup and recovery, complete the following steps.

1. Navigate to Protection > Backup and recovery > Virtual Machines and click on Virtual Machines to view the list of virtual machines available to be restored.

2. Access the settings drop down menu for the VM to be restored and select

3. Select the backup to restore from and click on Next.

4. Review a summary of the backup job and click on Restore to start the restore process.

5. Monitor the progress of the restore job from the Job Monitoring tab.
Conclusion

The 3-2-1 backup strategy, when implemented with SnapCenter Plug-in for VMware vSphere and BlueXP backup and recovery for virtual machines, offers a robust, reliable, and cost-effective solution for data protection. This strategy not only ensures data redundancy and accessibility but also provides the flexibility of restoring data from any location and from both on-premises ONTAP storage systems and cloud based object storage.

The use case presented in this documentation focuses on proven data protection technologies that highlight the integration between NetApp, VMware, and the leading cloud providers. The SnapCenter Plug-in for VMware vSphere provides seamless integration with VMware vSphere, allowing for efficient and centralized management of data protection operations. This integration streamlines the backup and recovery processes for virtual machines, enabling easy scheduling, monitoring, and flexible restore operations within the VMware ecosystem. BlueXP backup and recovery for virtual machines provides the one (1) in 3-2-1 by providing secure, air-gapped backups of virtual machine data to cloud based object storage. The intuitive interface and logical workflow provide a secure platform for long-term archival of critical data.

Additional Information

To learn more about the technologies presented in this solution refer to the following additional information.

- SnapCenter Plug-in for VMware vSphere documentation
- BlueXP documentation
VMware Sovereign Cloud

VMware Resources for Sovereign Cloud

NetApp and VMware Sovereign Cloud

Overview of VMware Sovereign Cloud

The concept of sovereignty is emerging as a necessary component of cloud computing for many entities that process and maintain highly sensitive data, such as national and state governments, and highly regulated industries, such as finance and healthcare. National governments are also looking to expand digital economic capability and reduce reliance on multi-national firms for their cloud services.

VMware Sovereign Cloud Initiative

VMware defines a sovereign cloud as one that:

- Protects and unlocks the value of critical data (e.g., national data, corporate data, and personal data) for both private and public sector organizations
- Delivers a national capability for the digital economy
- Secures data with audited security controls
- Ensures compliance with data privacy laws
- Improves control of data by providing both data residency and data sovereignty with full jurisdictional control

Partnering with a Trusted VMware Sovereign Cloud Service Provider

To ensure success, organizations must work with partners they trust and that are capable of hosting authentic and autonomous sovereign cloud platforms. VMware Cloud Providers recognized within the VMware Sovereign Cloud initiative commit to designing and operating cloud solutions based on modern, software-defined architectures that embody key principles and best practices outlined in the VMware Sovereign Cloud framework.

- **Data Sovereignty and Jurisdictional Control** – All data is resident and subject to the exclusive control and authority of the nation state where that data was collected. Operations are fully managed within the jurisdiction
- **Data Access and Integrity** – Cloud infrastructure is resilient and available in at least two data center locations within the jurisdiction with secure and private connectivity options available.
- **Data Security and Compliance** – Information security management system controls are certified against an industry recognized global (or regional) standard and audited regularly.
- **Data Independence and Mobility** – Support for modern application architectures to prevent vendor cloud lock-in and enable application portability and independence

For more information from VMware, please visit:

- VMware Sovereign Cloud Overview
Netpp with VMware Sovereign Cloud: Use Cases

NetApp provides support for VMware Sovereign Cloud concepts through the integration of several NetApp technologies.

Use the following link(s) to discover more about the NetApp technology integrations with VMware Sovereign Cloud:

• NetApp StorageGRID as an Object Store Extension

NetApp StorageGRID as an Object Store Extension

NetApp has collaborated with VMware to integrate NetApp StorageGRID into VMware Cloud Director in support of the VMware Sovereign Cloud. This plug-in to VMware Cloud Director enables service providers to use StorageGRID as their object storage offering (regardless of use case) and allows StorageGRID management through the same VMware multi-tenant solution (VMware Cloud Director) used by service providers to manage other parts of their offering catalog.

Partners that deliver VMware Sovereign Clouds can choose NetApp StorageGRID to help them managed and maintain cloud environments with unstructured data. Its universal compatibility in its native support for industry-standard APIs, like Amazon S3 API, helps ensure smooth interoperability across diverse cloud environments, and unique innovations such as automated lifecycle management helps ensure more cost-effective safeguarding, storage, and long-term preservation of customers’ unstructured data.

NetApp’s Sovereign Cloud integration with Cloud Director provides customers with:

• Assurance that sensitive data, including metadata, remains under sovereign control while preventing access by foreign authorities that could violate data privacy laws.

• Increased security and compliance that protects applications and data from rapidly evolving attack vectors while maintaining continuous compliance with a trusted local infrastructure, built-in frameworks, and local experts.

• Future-proofed infrastructure to react quickly to changing data privacy regulations, security threats, and geopolitics.

• The ability to unlock the value of data with secure data sharing and analysis to drive innovation without violating privacy laws. Data integrity is protected to ensure accurate insights.

For more information on the StorageGRID integration, check out the following:

• NetApp Announcement
NetApp Hybrid Multicloud with Red Hat OpenShift Container workloads

NetApp Hybrid Multicloud solutions for Red Hat OpenShift Container workloads

Overview

NetApp is seeing a significant increase in customers modernizing their legacy enterprise applications and building new applications using containers and orchestration platforms built around Kubernetes. Red Hat OpenShift Container Platform is one example that we see adopted by many of our customers.

As more and more customers begin adopting containers within their enterprises, NetApp is perfectly positioned to help serve the persistent storage needs of their stateful applications and classic data management needs such as data protection, data security, and data migration. However, these needs are met using different strategies, tools, and methods.

NetApp ONTAP based storage options listed below, deliver security, data protection, reliability, and flexibility for containers and Kubernetes deployments.

- Self-managed storage in on-premises:
  - NetApp Fabric Attached Storage (FAS), NetApp All Flash FAS Arrays (AFF), NetApp All SAN Array (ASA) and ONTAP Select
- Provider-managed storage in on-premises:
  - NetApp Keystone provides Storage as a Service (STaaS)
- Self-managed storage in the cloud:
  - NetApp Cloud Volumes ONTAP (CVO) provide self managed storage in the hyperscalers
- Provider-managed storage in the cloud:
  - Cloud Volumes Service for Google Cloud (CVS), Azure NetApp Files (ANF), Amazon FSx for NetApp ONTAP offer fully managed storage in the hyperscalers
NetApp BlueXP enables you to manage all of your storage and data assets from a single control plane/interface.

You can use BlueXP to create and administer cloud storage (for example, Cloud Volumes ONTAP and Azure NetApp Files), to move, protect, and analyze data, and to control many on-prem and edge storage devices.

NetApp Astra Trident is a CSI Compliant Storage Orchestrator that enable quick and easy consumption of persistent storage backed by a variety of the above-mentioned NetApp storage options. It is an open-source software maintained and supported by NetApp.

**Astra Trident CSI feature highlights**

<table>
<thead>
<tr>
<th>CSI specific</th>
<th>Security</th>
<th>Control</th>
<th>Installation methods</th>
<th>Choose your access mode</th>
<th>Choose your protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSI NetApp® Snapshot™ copies and volume creation from CSI Snapshot copies</td>
<td>Dynamic-export policy management</td>
<td>Storage and performance consumption</td>
<td>Binary</td>
<td>RWO (ReadWriteOnce, i.e. 1→1)</td>
<td>NFS</td>
</tr>
<tr>
<td>CSI topology</td>
<td>iSCSI initiator-groups dynamic management</td>
<td>Monitoring</td>
<td>Helm chart</td>
<td>RWX (ReadWriteMany, i.e. 1→n)</td>
<td>SMB</td>
</tr>
<tr>
<td>Volume expansion</td>
<td>iSCSI bidirectional CHAP</td>
<td></td>
<td>GitOps</td>
<td>ROX (ReadOnlyMany)</td>
<td>iSCSI</td>
</tr>
</tbody>
</table>
Business critical container workloads need more than just persistent volumes. Their data management requirements require protection and migration of the application Kubernetes objects as well.

Application data includes Kubernetes objects in addition to the user data: Some examples are as follows:
- Kubernetes objects such as pods specs, PVCs, deployments, services
- custom config objects such as config maps and secrets
- persistent data such as Snapshot copies, backups, clones
- custom resources such as CRs and CRDs

**NetApp Astra Control**, available as both fully-managed and self-managed software, provides orchestration for robust application data management. Refer to the Astra documentation for additional details on the Astra family of products.

This reference documentation provides validation of migration and protection of container-based applications, deployed on RedHat OpenShift container platform, using NetApp Astra Control Center. In addition, the solution provides high-level details for the deployment and the use of Red Hat Advanced Cluster Management (ACM) for managing the container platforms. The document also highlights the details for the integration of NetApp storage with Red Hat OpenShift container platforms using Astra Trident CSI provisioner. Astra Control Center is deployed on the hub cluster and is used to manage the container applications and their persistent storage lifecycle. Finally, it provides a solution for replication and failover and fail-back for container workloads on managed Red Hat OpenShift clusters in AWS (ROSA) using Amazon FSx for NetApp ONTAP (FSxN) as persistent storage.

**Value propositions of NetApp Hybrid Multicloud solutions for Red Hat OpenShift Container workloads**

Most customers do not just start out building Kubernetes based environments without any existing infrastructure. Perhaps they are a traditional IT shop running most of their enterprise applications on virtual machines (in large VMware environments for example). Then they start building small container-based environments to satisfy the needs of their modern application development teams. These initiatives usually start small and begin to become more pervasive as the teams learn these new technologies and skills, and begin to recognize the many benefits of adopting them.

The good news for customers is that NetApp can serve the needs of both environments. This set of solutions for hybrid multicloud with Red Hat OpenShift will empower NetApp customers to adopt modern cloud technologies and services without having to overhaul their entire infrastructure and organization. Whether customer applications and data are hosted on-premises, in cloud, run on virtual machines, or on containers, NetApp can provide consistent data management, protection, security, and portability. With these new solutions, the same value NetApp has delivered in on-premises data center environments for decades will be available across the enterprise entire data horizon, without requiring significant investment to retool, acquire new skills, or build new teams. NetApp is positioned well to help customers solve these business challenges regardless of what phase of their cloud journey they are in.

NetApp Hybrid Multi-Cloud with Red Hat OpenShift:

- Gives customers validated designs and practices which demonstrate the best ways for customers to manage, protect, secure, and migrate their data and applications when using Red Hat OpenShift with
NetApp based storage solutions.

• Present best practices for customers running Red Hat OpenShift with NetApp storage in VMware environments, bare metal infrastructure, or a combination of both.

• Demonstrate strategies and options for both on-prem and cloud environments, as well as hybrid environments where both are used.

**Supported Solutions of NetApp Hybrid Multicloud for Red Hat OpenShift Container workloads**

The solution tests and validates Migration & Centralized Data Protection with OpenShift container platform (OCP), OpenShift Advanced Cluster Manager (ACM), NetApp ONTAP, NetApp BlueXP and NetApp Astra Control Center (ACC).

For this solution, the following scenarios are tested and validated by NetApp. The solution is separated into multiple scenarios based on the following characteristics:

• on-premises

• cloud
  ◦ self-managed OpenShift clusters and self-managed NetApp storage
  ◦ provider-managed OpenShift clusters and provider-managed NetApp storage

*We will be building out additional solutions and use cases in the future.*

**Scenario 1: Data protection and migration within the on-premises environment using ACC**

**On-premises: self-managed OpenShift clusters and self-managed NetApp storage**

• Using ACC, create Snapshot copies, backups and restores for data protection.

• Using ACC, perform a SnapMirror replication of container applications.

**Scenario 1**
Scenario 2: Data protection and migration from the on-premises environment to AWS environment using ACC

On-premises: Self-managed OpenShift cluster and self-managed storage
AWS Cloud: Self-managed OpenShift cluster and self-managed storage

- Using ACC, perform backups and restores for data protection.
- Using ACC, perform a SnapMirror replication of container applications.

Scenario 2
Scenario 3: Data protection and migration from the on-premises environment to AWS environment

On-premises: Self-managed OpenShift cluster and self-managed storage
AWS Cloud: Provider-managed OpenShift cluster (ROSA) and provider-managed storage (FSxN)

- Using BlueXP, perform replication of persistent volumes (FSxN).
- Using OpenShift GitOps, recreate application metadata.

Scenario 3
Scenario 4: Data protection and migration from the on-premises environment to GCP environment using ACC

On-premises: Self-managed OpenShift cluster and self-managed storage
Google Cloud: Self-managed OpenShift cluster and self-managed storage

- Using ACC, perform backups and restores for data protection.
- Using ACC, perform a SnapMirror replication of container applications.
For considerations when using ONTAP in a MetroCluster configuration, refer here.

Scenario 5: Data protection and migration from the on-premises environment to Azure environment using ACC

On-premises: Self-managed OpenShift cluster and self-managed storage  
Azure Cloud: Self-managed OpenShift cluster and self-managed storage

- Using ACC, perform backups and restores for data protection.
- Using ACC, perform a SnapMirror replication of container applications.

---

For considerations when using ONTAP in a MetroCluster configuration, refer here.

### Versions of various components used in the solution validation

The solution tests and validates Migration & Centralized Data Protection with OpenShift container platform, OpenShift Advanced Cluster Manager, NetApp ONTAP, and NetApp Astra Control Center.

Scenarios 1, 2 and 3 of the solution were validated using the versions as shown in the table below:

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMware</td>
<td>vSphere Client version 8.0.0.10200</td>
</tr>
<tr>
<td></td>
<td>VMware ESXi, 8.0.0, 20842819</td>
</tr>
<tr>
<td>Hub Cluster</td>
<td>OpenShift 4.11.34</td>
</tr>
<tr>
<td>Source and Destination Clusters</td>
<td>OpenShift 4.12.9 on-premises and in AWS</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>NetApp Astra Trident</td>
<td>Trident Server and Client 23.04.0</td>
</tr>
<tr>
<td>NetApp Astra Control Center</td>
<td>ACC 22.11.0-82</td>
</tr>
<tr>
<td>NetApp ONTAP</td>
<td>ONTAP 9.12.1</td>
</tr>
<tr>
<td>AWS FSx for NetApp ONTAP</td>
<td>Single AZ</td>
</tr>
</tbody>
</table>

Scenario 4 of the solution was validated using the versions as shown in the table below:

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMware</td>
<td>vSphere Client version 8.0.2.00000</td>
</tr>
<tr>
<td></td>
<td>VMware ESXi, 8.0.2, 22380479</td>
</tr>
<tr>
<td>Hub Cluster</td>
<td>OpenShift 4.13.13</td>
</tr>
<tr>
<td>Source and Destination Clusters</td>
<td>OpenShift 4.13.12 on-premises and in Google Cloud</td>
</tr>
<tr>
<td>NetApp Astra Trident</td>
<td>Trident Server and Client 23.07.0</td>
</tr>
<tr>
<td>NetApp Astra Control Center</td>
<td>ACC 23.07.0-25</td>
</tr>
<tr>
<td>NetApp ONTAP</td>
<td>ONTAP 9.12.1</td>
</tr>
<tr>
<td>Cloud Volumes ONTAP</td>
<td>Single AZ, Single node, 9.14.0</td>
</tr>
</tbody>
</table>

Scenario 5 of the solution was validated using the versions as shown in the table below:

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMware</td>
<td>vSphere Client version 8.0.2.00000</td>
</tr>
<tr>
<td></td>
<td>VMware ESXi, 8.0.2, 22380479</td>
</tr>
<tr>
<td>Source and Destination Clusters</td>
<td>OpenShift 4.13.25 on-premises and in Azure</td>
</tr>
<tr>
<td>NetApp Astra Trident</td>
<td>Trident Server and Client and Astra Control Provisioner 23.10.0</td>
</tr>
<tr>
<td>NetApp Astra Control Center</td>
<td>ACC 23.10</td>
</tr>
<tr>
<td>NetApp ONTAP</td>
<td>ONTAP 9.12.1</td>
</tr>
<tr>
<td>Cloud Volumes ONTAP</td>
<td>Single AZ, Single node, 9.14.0</td>
</tr>
</tbody>
</table>

**Supported NetApp Storage integrations with Red Hat Open Shift Containers**

Whether the Red Hat Open Shift containers are running on VMware or in the hyperscalers, NetApp Astra Trident can be used as the CSI provisioner for the various types of backend NetApp storage that it supports.
The following diagram depicts the various backend NetApp storage that can be integrated with OpenShift clusters using NetApp Astra Trident.

ONTAP Storage Virtual Machine (SVM) provides secure multi-tenancy. A Single OpenShift cluster can connect to single SVM or multiple SVMs or even to multiple ONTAP clusters. Storage class filters the backend storage based on parameters or by labels. Storage administrators define the parameters to connect to storage system using trident backend configuration. On successful connection establishment, it creates the trident backend and populates the information which the storage class can filter.

The relationship between the storageclass and backend is shown below.
Application owner requests persistent volume using storage class. The storage class filters the backend storage.
The relationship between the pod and backend storage is shown below.
Container Storage Interface (CSI) Options

On vSphere environments, customers can pick VMware CSI driver and/or Astra Trident CSI to integrate with ONTAP. With VMware CSI, the persistent volumes are consumed as local SCSI disks, whereas with Trident, it is consumed with network. As VMware CSI does not support RWX access modes with ONTAP, applications need to use Trident CSI if RWX mode is required. With FC based deployments, VMware CSI is preferred and SnapMirror Business Continuity (SMBC) provides zone level high availability.
VMware CSI supports

- Core Block based datastores (FC, FCoE, iSCSI, NVMeoF)
- Core File based datastores (NFS v3, v4)
- vVol datastores (block and file)

Trident has following drivers to support ONTAP

- ontap-san (dedicated volume)
- ontap-san-economy (shared volume)
- ontap-nas (dedicated volume)
- ontap-nas-economy (shared volume)
- ontap-nas-flexgroup (dedicated large scale volume)

For both VMware CSI and Astra Trident CSI, ONTAP supports nconnect, session trunking, kerberos, etc. for NFS and multipathing, chap authentication, etc. for block protocols.

In AWS, FSx for NetApp ONTAP (FSxN) can be deployed in single Availability Zone (AZ) or in Multi AZ. For production workloads that requires high availability, multi-AZ provides zonal level fault tolerance and has better NVMe read cache compared to single AZ. For more info, check AWS performance guidelines.

To save cost on disaster recovery site, single AZ FSx ONTAP can be utilized.

For number of SVMs that are supported by FSx ONTAP, refer managing FSx ONTAP storage virtual machine

NetApp Hybrid Multicloud solutions for Red Hat OpenShift Container workloads
Overview

NetApp is seeing a significant increase in customers modernizing their legacy enterprise applications and building new applications using containers and orchestration platforms built around Kubernetes. Red Hat OpenShift Container Platform is one example that we see adopted by many of our customers.

As more and more customers begin adopting containers within their enterprises, NetApp is perfectly positioned to help serve the persistent storage needs of their stateful applications and classic data management needs such as data protection, data security, and data migration. However, these needs are met using different strategies, tools, and methods.

**NetApp ONTAP** based storage options listed below, deliver security, data protection, reliability, and flexibility for containers and Kubernetes deployments.

- Self-managed storage in on-premises:
  - NetApp Fabric Attached Storage (FAS), NetApp All Flash FAS Arrays (AFF), NetApp All SAN Array (ASA) and ONTAP Select
- Provider-managed storage in on-premises:
  - NetApp Keystone provides Storage as a Service (STaaS)
- Self-managed storage in the cloud:
  - NetApp Cloud Volumes ONTAP (CVO) provide self managed storage in the hyperscalers
- Provider-managed storage in the cloud:
  - Cloud Volumes Service for Google Cloud (CVS), Azure NetApp Files (ANF), Amazon FSx for NetApp ONTAP offer fully managed storage in the hyperscalers

**ONTAP feature highlights**

<table>
<thead>
<tr>
<th>Storage Administration</th>
<th>Performance &amp; Scalability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-tenancy</td>
<td>FlexCache</td>
</tr>
<tr>
<td>FlexVol &amp; FlexGroup</td>
<td>FlexClone</td>
</tr>
<tr>
<td>LUN</td>
<td>Scale-out clusters</td>
</tr>
<tr>
<td>Quotas</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ONTAP CLI &amp; API</td>
</tr>
<tr>
<td></td>
<td>System Manager &amp; BlueXP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Availability &amp; Resilience</th>
<th>Access Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-AZ HA deployment (MetroCluster)</td>
<td>NFS – v3, v4, v4.1, v4.2</td>
</tr>
<tr>
<td>SnapShot &amp; SnapRestore</td>
<td>ISCSI</td>
</tr>
<tr>
<td>SnapMirror</td>
<td>Multi-protocol access</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Storage Efficiency</th>
<th>Security &amp; Compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deduplication &amp; Compression</td>
<td>FPolicy &amp; Vscan</td>
</tr>
<tr>
<td>Compaction</td>
<td>LDAP &amp; Kerberos</td>
</tr>
<tr>
<td>Thin provisioning</td>
<td>Certificate based authentication</td>
</tr>
<tr>
<td>Data Tiering (Fabric Pool)</td>
<td></td>
</tr>
</tbody>
</table>

**NetApp BlueXP** enables you to manage all of your storage and data assets from a single control plane/interface.

You can use BlueXP to create and administer cloud storage (for example, Cloud Volumes ONTAP and Azure
NetApp Files), to move, protect, and analyze data, and to control many on-prem and edge storage devices.

**NetApp Astra Trident** is a CSI Compliant Storage Orchestrator that enables quick and easy consumption of persistent storage backed by a variety of the above-mentioned NetApp storage options. It is an open-source software maintained and supported by NetApp.

### Astra Trident CSI feature highlights

<table>
<thead>
<tr>
<th><strong>CSI specific</strong></th>
<th><strong>Security</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>- CSI NetApp® Snapshot™ copies and volume creation from CSI Snapshot copies</td>
<td>- Dynamic-export policy management</td>
</tr>
<tr>
<td>- CSI topology</td>
<td>- iSCSI initiator-groups dynamic management</td>
</tr>
<tr>
<td>- Volume expansion</td>
<td>- iSCSI bidirectional CHAP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Control</strong></th>
<th><strong>Installation methods</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>- Storage and performance consumption</td>
<td>- Binary</td>
</tr>
<tr>
<td>- Monitoring</td>
<td>- Helm chart</td>
</tr>
<tr>
<td></td>
<td>- Operator</td>
</tr>
<tr>
<td></td>
<td>- GitOps</td>
</tr>
<tr>
<td></td>
<td>- Cross Namespace Volume Access</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Choose your access mode</strong></th>
<th><strong>Choose your protocol</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>- RWO (ReadWriteOnce, i.e 1:1)</td>
<td>- NFS</td>
</tr>
<tr>
<td>- RWX (ReadWriteMany, i.e 1:n)</td>
<td>- SMB</td>
</tr>
<tr>
<td>- ROX (ReadOnlyMany)</td>
<td>- iSCSI</td>
</tr>
</tbody>
</table>

Business critical container workloads need more than just persistent volumes. Their data management requirements require protection and migration of the application Kubernetes objects as well.

Application data includes Kubernetes objects in addition to the user data: Some examples are as follows:
- Kubernetes objects such as pods specs, PVCs, deployments, services
- Custom config objects such as config maps and secrets
- Persistent data such as Snapshot copies, backups, clones
- Custom resources such as CRs and CRDs

**NetApp Astra Control**, available as both fully-managed and self-managed software, provides orchestration for robust application data management. Refer to the [Astra documentation](#) for additional details on the Astra family of products.

This reference documentation provides validation of migration and protection of container-based applications, deployed on RedHat OpenShift container platform, using NetApp Astra Control Center. In addition, the solution provides high-level details for the deployment and the use of Red Hat Advanced Cluster Management (ACM) for managing the container platforms. The document also highlights the details for the integration of NetApp storage with Red Hat OpenShift container platforms using Astra Trident CSI provisioner. Astra Control Center is deployed on the hub cluster and is used to manage the container applications and their persistent storage lifecycle. Finally, it provides a solution for replication and failover and fail-back for container workloads on managed Red Hat OpenShift clusters in AWS (ROSA) using Amazon FSx for NetApp ONTAP (FSxN) as persistent storage.
NetApp Solution with Red Hat OpenShift Container platform workloads on VMware

If customers have a need to run their modern containerized applications on infrastructure in their private data centers, they can do so. They should plan and deploy the Red Hat OpenShift container platform (OCP) for a successful production-ready environment for deploying their container workloads. Their OCP clusters can be deployed on VMware or bare metal.

NetApp ONTAP storage delivers data protection, reliability, and flexibility for container deployments. Astra Trident serves as the dynamic storage provisioner to consume persistent ONTAP storage for customers’ stateful applications. Astra Control Center can be used to orchestrate the many data management requirements of stateful applications such as data protection, migration, and business continuity.

With VMware vSphere, NetApp ONTAP tools provides a vCenter Plugin which can be utilized to provision datastores. Apply tags and use it with OpenShift for storing the node configuration and data. NVMe based storage provides lower latency and high performance.

This solution provides details for data protection and migration of container workloads using Astra Control Center. For this solution, the container workloads are deployed on Red Hat OpenShift clusters on vSphere within the on-premises environment.

NOTE: We will provide a solution for container workloads on OpenShift clusters on bare metal in the future.

Data protection and migration solution for OpenShift Container workloads using Astra Control Center

Deploy and configure the Red Hat OpenShift Container platform on VMware

This section describes a high-level workflow of how to set up and manage OpenShift clusters and manage stateful applications on them. It shows the use of NetApp ONTAP storage arrays with the help of Astra Trident to provide persistent volumes. Details are
provided about the use of Astra Control Center to perform data protection and migration activities for the stateful applications.

There are several ways of deploying Red Hat OpenShift Container platform clusters. This high-level description of the setup provides documentation links for the specific method that was used. You can refer to the other methods in the relevant links provided in the resources section.

Here is a diagram that depicts the clusters deployed on VMware in a data center.

The setup process can be broken down into the following steps:

**Deploy and configure a CentOS VM**

- It is deployed in the VMware vSphere environment.
- This VM is used for deploying some components such as NetApp Astra Trident and NetApp Astra Control Center for the solution.
- A root user is configured on this VM during installation.
Deploy and configure an OpenShift Container Platform cluster on VMware vSphere (Hub Cluster)

Refer to the instructions for the Assisted deployment method to deploy an OCP cluster.

Remember the following:
- Create ssh public and private key to provide to the installer. These keys will be used to login to the master and worker nodes if needed.
- Download the installer program from the assisted installer. This program is used to boot the VMs that you create in the VMware vSphere environment for the master and worker nodes.
- VMs should have the minimum CPU, memory, and hard disk requirement. (Refer to the `vm create` commands on this page for the master and the worker nodes which provide this information)
- The diskUUID should be enabled on all VMs.
- Create a minimum of 3 nodes for master and 3 nodes for worker.
- Once they are discovered by the installer, turn on the VMware vSphere integration toggle button.

Install Advanced Cluster Management on the Hub cluster

This is installed using the Advanced Cluster Management Operator on the Hub Cluster. Refer to the instructions [here](#).

Install an internal Red Hat Quay registry on the Hub Cluster.

- An internal registry is required to push the Astra image. A Quay internal registry is installed using the Operator in the Hub cluster.
- Refer to the instructions [here](#)

Install two additional OCP clusters (Source and Destination)

- The additional clusters can be deployed using the ACM on the Hub Cluster.
- Refer to the instructions [here](#)

Configure NetApp ONTAP storage

- Install an ONTAP cluster with connectivity to the OCP VMs in VMWare environment.
- Create an SVM.
- Configure NAS data lif to access the storage in SVM.
Install NetApp Trident on the OCP clusters

- Install NetApp Trident on all three clusters: Hub, source, and destination clusters
- Refer to the instructions here.
- Create a storage backend for ontap-nas.
- Create a storage class for ontap-nas.
- Refer to instructions here.

Install NetApp Astra Control Center

- NetApp Astra Control Center is installed using the Astra Operator on the Hub Cluster.
- Refer to the instructions here.

Points to remember:
* Download NetApp Astra Control Center image from the support site.
* Push the image to an internal registry.
* Refer to instructions here.

Deploy an Application on Source Cluster

Use OpenShift GitOps to deploy an application. (eg. Postgres, Ghost)

Add the Source and Destination clusters into Astra Control Center.

After you add a cluster to Astra Control management, you can install apps on the cluster (outside of Astra Control) and then go to the Applications page in Astra Control to define the apps and their resources. Refer to Start managing apps section of Astra Control Center.

The next step is to use the Astra Control Center for Data protection and Data migration from the source to the destination cluster.

Data protection using Astra

This page shows the data protection options for Red Hat OpenShift Container based applications running on VMware vSphere using Astra Control Center (ACC).

As users take their journey of modernizing their applications with Red Hat OpenShift, a data protection strategy should be in place to protect them from accidental deletion or any other human errors. Often a protection strategy is also required for regulatory or compliance purposes to protect their data from a disaster.

The requirements of data protection varies from reverting back to a point in time copy to automatically failing over to a different fault domain without any human intervention. Many customers pick ONTAP as their preferred storage platform for their Kubernetes applications because of its rich features like multitenancy, multi-protocol, high performance and capacity offerings, replication and caching for multi-site locations, security and flexibility.

Data protection in ONTAP can be achieved using ad-hoc or policy controlled - Snapshot
- **backup and restore**

Both Snapshot copies and backups protect the following types of data:
- The application metadata that represents the state of the application
- Any persistent data volumes associated with the application
- Any resource artifacts belonging to the application

**Snapshot with ACC**

A point in time copy of data can be captured using Snapshot with ACC. Protection policy defines the number of Snapshot copies to keep. Minimum schedule option available is hourly. Manual, on-demand Snapshot copies can be taken at any time and at shorter intervals than scheduled Snapshot copies. Snapshot copies are stored on the same provisioned volume as the app.

**Configuring Snapshot with ACC**

![Snapshot with ACC configuration](image)

**Backup and Restore with ACC**

A backup is based on a Snapshot. ACC can take Snapshot copies using CSI and perform backup using the point in time Snapshot copy. The backup is stored in an external object store (any s3 compatible including ONTAP S3 at a different location). Protection policy can be configured for scheduled backups and the number of backup versions to keep. The minimum RPO is one hour.

**Restoring an application from a backup using ACC**

ACC restores application from the S3 bucket where the backups are store.

![Backup and Restore with ACC](image)
Application specific execution hooks

In addition, execution hooks can be configured to run in conjunction with a data protection operation of a managed app. Even though storage array level data protection features are available, often additional steps are needed to make backups and restores, application consistent. The app-specific additional steps could be:
- before or after a Snapshot copy is created.
- before or after a backup is created.
- after restoring from a Snapshot copy or backup.

Astra Control can execute these app-specific steps coded as custom scripts called execution hooks.

NetApp Verda GitHub project provides execution hooks for popular cloud-native applications to make protecting applications straightforward, robust, and easy to orchestrate. Feel free to contribute to that project if you have enough information for an application that is not in the repository.

Sample execution hook for pre-Snapshot of a redis application.

Replication with ACC

For regional protection or for a low RPO and RTO solution, an application can be replicated to another Kubernetes instance running at a different site, preferably in another region. ACC utilizes ONTAP async SnapMirror with RPO as low as 5 minutes. Replication is done by replicating to ONTAP and then a fail over
creates the Kubernetes resources in the destination cluster.

Note that replication is different from the backup and restore where the backup goes to S3 and restore is performed from S3. Refer xref:/rhhc/ here to get additional details about the differences between the two types of data protection.

Refer here for SnapMirror setup instructions.

SnapMirror with ACC

San-economy and nas-economy storage drivers do not support replication feature. Refer here for additional details.

Demo video:

Demonstration video of disaster recovery with Astra Control Center

Data protection with Astra Control Center

Business Continuity with MetroCluster

Most of our hardware platform for ONTAP has high availability features to protect from device failures avoiding the need to perform disaster recovery. But to protect from fire or any other disaster and to continue the business with zero RPO and low RTO, often a MetroCluster solution is used.

Customers who currently have an ONTAP system can extend to MetroCluster by adding supported ONTAP systems within the distance limitations for providing zone level disaster recovery. Astra Trident, the CSI (Container Storage Interface) supports NetApp ONTAP including MetroCluster configuration as well as other options like Cloud Volumes ONTAP, Azure NetApp Files, AWS FSx for NetApp ONTAP, etc. Astra Trident provides five storage driver options for ONTAP and all are supported for MetroCluster configuration. Refer here for additional details about ONTAP storage drivers supported by Astra
Trident.

The MetroCluster solution requires layer 2 network extension or capability to access the same network address from both fault domains. Once MetroCluster configuration is in place, the solution is transparent to application owners as all the volumes in the MetroCluster svm are protected and get the benefits of SyncMirror (zero RPO).

For Trident Backend Configuration (TBC), do not specify the dataLIF and SVM when using MetroCluster configuration. Specify SVM management IP for managementLIF and use vsadmin role credentials.

Details on Astra Control Center Data Protection features are available here.

Data migration using Astra Control Center

This page shows the data migration options for container workloads on Red Hat OpenShift clusters with Astra Control Center (ACC).

Kubernetes Applications are often required to be moved from one environment to another. To migrate an application along with its persistent data, NetApp ACC can be utilized.

Data Migration between different Kubernetes environment

ACC supports various Kubernetes flavors including Google Anthos, Red Hat OpenShift, Tanzu Kubernetes Grid, Rancher Kubernetes Engine, Upstream Kubernetes, etc. For additional details, refer here.
To migrate an application from one cluster to another, you can use one of the following features of ACC:

- replication
- backup and restore
- clone

Refer to the data protection section for the replication and backup and restore options.

Refer here for additional details about cloning.

Astra Replication feature is only supported with Trident Container Storage Interface (CSI). However, replication is not supported by nas-economy & san-economy drivers.

Performing data replication using ACC

NetApp Hybrid Multicloud solutions for Red Hat OpenShift Container workloads

Overview

NetApp is seeing a significant increase in customers modernizing their legacy enterprise applications and building new applications using containers and orchestration platforms built around Kubernetes. Red Hat OpenShift Container Platform is one example that we see adopted by many of our customers.

As more and more customers begin adopting containers within their enterprises, NetApp is perfectly positioned to help serve the persistent storage needs of their stateful applications and classic data management needs such as data protection, data security, and data migration. However, these needs are met using different strategies, tools, and methods.
NetApp ONTAP based storage options listed below, deliver security, data protection, reliability, and flexibility for containers and Kubernetes deployments.

- **Self-managed storage in on-premises:**
  - NetApp Fabric Attached Storage (FAS), NetApp All Flash FAS Arrays (AFF), NetApp All SAN Array (ASA) and ONTAP Select

- **Provider-managed storage in on-premises:**
  - NetApp Keystone provides Storage as a Service (STaaS)

- **Self-managed storage in the cloud:**
  - NetApp Cloud Volumes ONTAP(CVO) provide self managed storage in the hyperscalers

- **Provider-managed storage in the cloud:**
  - Cloud Volumes Service for Google Cloud (CVS), Azure NetApp Files (ANF), Amazon FSx for NetApp ONTAP offer fully managed storage in the hyperscalers

### ONTAP feature highlights

<table>
<thead>
<tr>
<th>Storage Administration</th>
<th>Performance &amp; Scalability</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Multi-tenancy</td>
<td>- ONTAP CLI &amp; API</td>
</tr>
<tr>
<td>- FlexVol &amp; FlexGroup</td>
<td>- System Manager &amp; BlueXP</td>
</tr>
<tr>
<td>- LUN</td>
<td></td>
</tr>
<tr>
<td>- Quotas</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Availability &amp; Resilience</th>
<th>Access Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Multi-AZ HA deployment (MetroCluster)</td>
<td>- NFS –v3, v4, v4.1, v4.2</td>
</tr>
<tr>
<td>- SnapShot &amp; SnapRestore</td>
<td>- iSCSI</td>
</tr>
<tr>
<td>- SnapMirror</td>
<td>- Multi-protocol access</td>
</tr>
<tr>
<td>- SnapMirror</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Storage Efficiency</th>
<th>Security &amp; Compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Deduplication &amp; Compression</td>
<td>- Fpolicy &amp; Vscan</td>
</tr>
<tr>
<td>- Compaction</td>
<td>- Data Tiering (Fabric Pool)</td>
</tr>
<tr>
<td></td>
<td>- LDAP &amp; Kerberos</td>
</tr>
<tr>
<td></td>
<td>- Certificate based authentication</td>
</tr>
</tbody>
</table>

**NetApp BlueXP** enables you to manage all of your storage and data assets from a single control plane/interface.

You can use BlueXP to create and administer cloud storage (for example, Cloud Volumes ONTAP and Azure NetApp Files), to move, protect, and analyze data, and to control many on-prem and edge storage devices.

**NetApp Astra Trident** is a CSI Compliant Storage Orchestrator that enable quick and easy consumption of persistent storage backed by a variety of the above-mentioned NetApp storage options. It is an open-source software maintained and supported by NetApp.
Business critical container workloads need more than just persistent volumes. Their data management requirements require protection and migration of the application Kubernetes objects as well.

Application data includes Kubernetes objects in addition to the user data: Some examples are as follows:
- Kubernetes objects such as pods specs, PVCs, deployments, services
- Custom config objects such as config maps and secrets
- Persistent data such as Snapshot copies, backups, clones
- Custom resources such as CRs and CRDs

**NetApp Astra Control**, available as both fully-managed and self-managed software, provides orchestration for robust application data management. Refer to the [Astra documentation](#) for additional details on the Astra family of products.

This reference documentation provides validation of migration and protection of container-based applications, deployed on Red Hat OpenShift container platform, using NetApp Astra Control Center. In addition, the solution provides high-level details for the deployment and the use of Red Hat Advanced Cluster Management (ACM) for managing the container platforms. The document also highlights the details for the integration of NetApp storage with Red Hat OpenShift container platforms using Astra Trident CSI provisioner. Astra Control Center is deployed on the hub cluster and is used to manage the container applications and their persistent storage lifecycle. Finally, it provides a solution for replication and failover and fail-back for container workloads on managed Red Hat OpenShift clusters in AWS (ROSA) using Amazon FSx for NetApp ONTAP (FSxN) as persistent storage.

**NetApp Solution with Red Hat OpenShift Container platform workloads in Hybrid Cloud**

Customers may be at a point in their modernization journey when they are ready to move some select workloads or all workloads from their data centers to the cloud. They may choose to use self-managed OpenShift containers and self-managed NetApp storage in the cloud for various reasons. They should plan and deploy the Red Hat OpenShift
container platform (OCP) in the cloud for a successful production-ready environment for migrating their container workloads from their data centers. Their OCP clusters can be deployed on VMware or Bare Metal in their data centers and on AWS, Azure or Google Cloud in the cloud environment.

NetApp Cloud Volumes ONTAP storage delivers data protection, reliability, and flexibility for container deployments in AWS, Azure and in Google Cloud. Astra Trident serves as the dynamic storage provisioner to consume the persistent Cloud Volumes ONTAP storage for customers’ stateful applications. Astra Control Center can be used to orchestrate the many data management requirements of stateful applications such as data protection, migration, and business continuity.

Data protection and migration solution for OpenShift Container workloads in a hybrid cloud using Astra Control Center

On-premises and AWS

On-premises and Google Cloud
On-premises and Azure Cloud
Deploy and configure the Red Hat OpenShift Container platform on AWS

This section describes a high-level workflow of how to set up and manage OpenShift Clusters in AWS and deploy stateful applications on them. It shows the use of NetApp Cloud Volumes ONTAP storage with the help of Astra Trident to provide persistent volumes. Details are provided about the use of Astra Control Center to perform data protection and migration activities for the stateful applications.

There are several ways of deploying Red Hat OpenShift Container platform clusters on AWS. This high-level description of the setup provides documentation links for the specific method that was used. You can refer to the other methods in the relevant links provided in the resources section.

Here is a diagram that depicts the clusters deployed on AWS and connected to the data center using a VPN.

The setup process can be broken down into the following steps:
Install an OCP cluster on AWS from the Advanced Cluster Management.

- Create a VPC with a site-to-site VPN connection (using pfsense) to connect to the on-premises network.
- On-premises network has internet connectivity.
- Create 3 private subnets in 3 different AZs.
- Create a Route 53 private hosted zone and a DNS resolver for the VPC.

Create OpenShift Cluster on AWS from the Advanced Cluster Management (ACM) Wizard. Refer to instructions here.

You can also create the cluster in AWS from the OpenShift Hybrid Cloud console. Refer here for instructions.

When creating the cluster using the ACM, you have the ability to customize the installation by editing the yaml file after filling in the details in the form view. After the cluster is created, you can ssh login to the nodes of the cluster for troubleshooting or additional manual configuration. Use the ssh key you provided during installation and the username core to login.

Deploy Cloud Volumes ONTAP in AWS using BlueXP.

- Install the connector in on-premises VMware environment. Refer to instructions here.
- Deploy a CVO instance in AWS using the connector. Refer to instructions here.

The connector can also be installed in the cloud environment. Refer here for additional information.

Install Astra Trident in the OCP Cluster

- Deploy Trident Operator using Helm. Refer to instructions here
- Create a backend and a storage class. Refer to instructions here.

Add the OCP cluster on AWS to the Astra Control Center.

Add the OCP cluster in AWS to Astra Control Center.

Using CSI Topology feature of Trident for multi-zone architectures

Cloud providers, today, enable Kubernetes/OpenShift cluster administrators to spawn nodes of the clusters that are zone based. Nodes can be located in different availability zones within a region, or across various regions. To facilitate the provisioning of volumes for workloads in a multi-zone architecture, Astra Trident uses CSI Topology. Using the CSI Topology feature, access to volumes can be limited to a subset of nodes, based on regions and availability zones. Refer here for additional details.
Kubernetes supports two volume binding modes:
- When `VolumeBindingMode` is set to `Immediate` (default), Astra Trident creates the volume without any topology awareness. Persistent Volumes are created without having any dependency on the requesting pod’s scheduling requirements.
- When `VolumeBindingMode` set to `WaitForFirstConsumer`, the creation and binding of a Persistent Volume for a PVC is delayed until a pod that uses the PVC is scheduled and created. This way, volumes are created to meet the scheduling constraints that are enforced by topology requirements.

Astra Trident storage backends can be designed to selectively provision volumes based on availability zones (Topology-aware backend). For StorageClasses that make use of such a backend, a volume would only be created if requested by an application that is scheduled in a supported region/zone. (Topology-aware StorageClass)

Refer here for additional details.

**Deploy and configure the Red Hat OpenShift Container platform on GCP**

This section describes a high-level workflow of how to set up and manage OpenShift Clusters in GCP and deploy stateful applications on them. It shows the use of NetApp Cloud Volumes ONTAP storage with the help of Astra Trident to provide persistent volumes. Details are provided about the use of Astra Control Center to perform data protection and migration activities for the stateful applications.

Here is a diagram that shows the clusters deployed on GCP and connected to the data center using a VPN.

There are several ways of deploying Red Hat OpenShift Container platform clusters in GCP. This high-level description of the setup provides documentation links for the specific method that was used. You can refer to the other methods in the relevant links provided in the resources section.
The setup process can be broken down into the following steps:

**Install an OCP cluster on GCP from the CLI.**

- Ensure that you have met all the prerequisites stated [here](#).
- For the VPN connectivity between on-premises and GCP, a pfense VM was created and configured. For instructions, see [here](#).
  - The remote gateway address in pfense can be configured only after you have created a VPN gateway in Google Cloud Platform.
  - The remote network IP addresses for the Phase 2 can be configured only after the OpenShift cluster installation program runs and creates the infrastructure components for the cluster.
  - The VPN in Google Cloud can only be configured after the infrastructure components for the cluster are created by the installation program.
- Now install the OpenShift cluster on GCP.
  - Obtain the installation program and the pull secret and deploy the cluster following the steps provided in the documentation [here](#).
  - The installation creates a VPC network in Google Cloud Platform. It also creates a private zone in Cloud DNS and adds A records.
    - Use the CIDR block address of the VPC network to configure the pfense and establish the VPN connection. Ensure firewalls are setup correctly.
    - Add A records in the DNS of the on-premises environment using the IP address in the A records of the Google Cloud DNS.
  - The installation of the cluster completes and will provide a kubeconfig file and username and password to login to the console of the cluster.

**Deploy Cloud Volumes ONTAP in GCP using BlueXP.**

- Install a connector in Google Cloud. Refer to instructions [here](#).
- Deploy a CVO instance in Google Cloud using the connector. Refer to instructions [here](#).
  
**Install Astra Trident in the OCP Cluster in GCP**

- There are many methods to deploy Astra Trident as shown [here](#).
- For this project, Astra Trident was installed by deploying Astra Trident Operator manually using the instructions [here](#).
- Create backend and a storage classes. Refer to instructions [here](#).
Add the OCP cluster on GCP to the Astra Control Center.

- Create a separate KubeConfig file with a cluster role that contains the minimum permissions necessary for a cluster to be managed by Astra Control. The instructions can be found here.
- Add the cluster to Astra Control Center following the instructions here

Using CSI Topology feature of Trident for multi-zone architectures

Cloud providers, today, enable Kubernetes/OpenShift cluster administrators to spawn nodes of the clusters that are zone based. Nodes can be located in different availability zones within a region, or across various regions. To facilitate the provisioning of volumes for workloads in a multi-zone architecture, Astra Trident uses CSI Topology. Using the CSI Topology feature, access to volumes can be limited to a subset of nodes, based on regions and availability zones. Refer here for additional details.

Kubernetes supports two volume binding modes:
- When **VolumeBindingMode** is set to **Immediate** (default), Astra Trident creates the volume without any topology awareness. Persistent Volumes are created without having any dependency on the requesting pod’s scheduling requirements.
- When **VolumeBindingMode** set to **WaitForFirstConsumer**, the creation and binding of a Persistent Volume for a PVC is delayed until a pod that uses the PVC is scheduled and created. This way, volumes are created to meet the scheduling constraints that are enforced by topology requirements.

Astra Trident storage backends can be designed to selectively provision volumes based on availability zones (Topology-aware backend). For StorageClasses that make use of such a backend, a volume would only be created if requested by an application that is scheduled in a supported region/zone. (Topology-aware StorageClass) Refer here for additional details.

Demonstration Video

OpenShift Cluster installation on Google Cloud Platform

Importing OpenShift clusters into Astra Control Center

Deploy and configure the Red Hat OpenShift Container platform on Azure

This section describes a high-level workflow of how to set up and manage OpenShift Clusters in Azure and deploy stateful applications on them. It shows the use of NetApp Cloud Volumes ONTAP storage with the help of Astra Trident/Astra Control Provisioner to provide persistent volumes. Details are provided about the use of Astra Control Center to perform data protection and migration activities for the stateful applications.

Here is a diagram that shows the clusters deployed on Azure and connected to the data center using a VPN.
There are several ways of deploying Red Hat OpenShift Container platform clusters in Azure. This high-level description of the setup provides documentation links for the specific method that was used. You can refer to the other methods in the relevant links provided in the resources section.

The setup process can be broken down into the following steps:
Install an OCP cluster on Azure from the CLI.

- Ensure that you have met all the prerequisites stated here.
- Create a VPN, subnets and network security groups and a private DNS zone. Create VPN gateway and site-to-site VPN Connection.
- For the VPN connectivity between on-premises and Azure, a pfense VM was created and configured. For instructions, see here.
- Obtain the installation program and the pull secret and deploy the cluster following the steps provided in the documentation here.
- The installation of the cluster completes and will provide a kubeconfig file and username and password to login to the console of the cluster.

A sample install-config.yaml file is given below.

```yaml
apiVersion: v1
baseDomain: sddc.netapp.com
compute:
  - architecture: amd64
    hyperthreading: Enabled
    name: worker
    platform:
      azure:
        encryptionAtHost: false
        osDisk:
          diskSizeGB: 512
          diskType: "StandardSSD_LRS"
          type: Standard_D2s_v3
          ultraSSDCapability: Disabled
# zones:
#  - "1"
#  - "2"
#  - "3"
replicas: 3
controlPlane:
  architecture: amd64
  hyperthreading: Enabled
  name: master
  platform:
    azure:
      encryptionAtHost: false
      osDisk:
        diskSizeGB: 1024
        diskType: Premium_LRS
        type: Standard_D8s_v3
        ultraSSDCapability: Disabled
      replicas: 3
```
metadata:
    creationTimestamp: null
    name: azure-cluster
networking:
    clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
    machineNetwork:
    - cidr: 10.0.0.0/16
      networkType: OVNKubernetes
    serviceNetwork:
    - 172.30.0.0/16
platform:
    azure:
        baseDomainResourceGroupName: ocp-base-domain-rg
        cloudName: AzurePublicCloud
        computeSubnet: ocp-subnet2
        controlPlaneSubnet: ocp-subnet1
        defaultMachinePlatform:
            osDisk:
                diskSizeGB: 1024
                diskType: "StandardSSD_LRS"
                ultraSSDCapability: Disabled
            networkResourceGroupName: ocp-nc-us-rg
        region: northcentralus
        resourceGroupName: ocp-cluster-ncusrg
        virtualNetwork: ocp_vnet_ncus
    publish: Internal
    pullSecret:

Deploy Cloud Volumes ONTAP in Azure using BlueXP.

- Deploy a CVO instance in Azure using the connector. Refer to instructions link:https://docs.netapp.com/us-en/bluexp-cloud-volumes-ontap/task-getting-started-azure.html [here.]

Install Astra Control Provisioner in the OCP Cluster in Azure

- For this project, Astra Control Provisioner (ACP) was installed on all the clusters (on-prem cluster, on-prem cluster where Astra Control Center is deployed and the cluster in Azure). Learn more about the Astra Control Provisioner [here](https://docs.netapp.com/us-en/bluexp-cloud-volumes-ontap/task-getting-started-azure.html).
- Create backend and a storage classes. Refer to instructions [here](https://docs.netapp.com/us-en/bluexp-cloud-volumes-ontap/task-getting-started-azure.html).
Add the OCP cluster on Azure to the Astra Control Center.

- Create a separate KubeConfig file with a cluster role that contains the minimum permissions necessary for a cluster to be managed by Astra Control. The instructions can be found here.
- Add the cluster to Astra Control Center following the instructions here

Using CSI Topology feature of Trident for multi-zone architectures

Cloud providers, today, enable Kubernetes/OpenShift cluster administrators to spawn nodes of the clusters that are zone based. Nodes can be located in different availability zones within a region, or across various regions. To facilitate the provisioning of volumes for workloads in a multi-zone architecture, Astra Trident uses CSI Topology. Using the CSI Topology feature, access to volumes can be limited to a subset of nodes, based on regions and availability zones. Refer here for additional details.

Kubernetes supports two volume binding modes:
- When **VolumeBindingMode is set to Immediate** (default), Astra Trident creates the volume without any topology awareness. Persistent Volumes are created without having any dependency on the requesting pod’s scheduling requirements.
- When **VolumeBindingMode set to WaitForFirstConsumer**, the creation and binding of a Persistent Volume for a PVC is delayed until a pod that uses the PVC is scheduled and created. This way, volumes are created to meet the scheduling constraints that are enforced by topology requirements.

Astra Trident storage backends can be designed to selectively provision volumes based on availability zones (Topology-aware backend). For StorageClasses that make use of such a backend, a volume would only be created if requested by an application that is scheduled in a supported region/zone. (Topology-aware StorageClass) Refer here for additional details.

Demonstration Video

Using Astra Control for Failover and Failback of applications

Data protection using Astra Control Center

This page shows the data protection options for Red Hat OpenShift Container based applications running on VMware vSphere or in the cloud using Astra Control Center (ACC).

As users take their journey of modernizing their applications with Red Hat OpenShift, a data protection strategy should be in place to protect them from accidental deletion or any other human errors. Often a protection strategy is also required for regulatory or compliance purposes to protect their data from a disaster.

The requirements of data protection varies from reverting back to a point in time copy to automatically failing over to a different fault domain without any human intervention. Many customers pick ONTAP as their preferred storage platform for their Kubernetes applications because of its rich features like multitenancy, multi-protocol, high performance and capacity offerings, replication and caching for multi-site locations, security and flexibility.

Customers may have a cloud environment setup as their data center extension, so that they can leverage the benefits of the cloud as well as be well positioned to move their workloads at a future time. For such customers, backing up of their OpenShift applications and their data to the cloud environment becomes an
inevitable choice. They can then restore the applications and the associated data either to an OpenShift cluster in the cloud or in their data center.

Backup and Restore with ACC

Application owners can review and update the applications discovered by ACC. ACC can take Snapshot copies using CSI and perform backup using the point in time Snapshot copy. Backup destination can be an object store in the cloud environment. Protection policy can be configured for scheduled backups and the number of backup versions to keep. The minimum RPO is one hour.

Restoring an application from a backup using ACC

Application specific execution hooks

Even though storage array level data protection features are available, often additional steps are needed to make backups and restores application consistent. The app-specific additional steps could be:
- before or after a Snapshot copy is created.
- before or after a backup is created.
- after restoring from a Snapshot copy or backup.
Astra Control can execute these app-specific steps coded as custom scripts called execution hooks.

NetApp’s open source project Verda provides execution hooks for popular cloud-native applications to make protecting applications straightforward, robust, and easy to orchestrate. Feel free to contribute to that project if you have enough information for an application that is not in the repository.

Sample execution hook for pre-Snapshot of a redis application.
Replication with ACC

For regional protection or for a low RPO and RTO solution, an application can be replicated to another Kubernetes instance running at a different site, preferably in another region. ACC utilizes ONTAP async SnapMirror with RPO as low as 5 minutes. Refer here for SnapMirror setup instructions.

SnapMirror with ACC
san-economy and nas-economy storage drivers do not support replication feature. Refer here for additional details.

Demo video:

Demonstration video of disaster recovery with Astra Control Center

Data protection with Astra Control Center

Details on Astra Control Center Data Protection features are available here

Disaster recovery (Failover and Failback using replication) with ACC

Using Astra Control for Failover and Failback of applications

Data migration using Astra Control Center

This page shows the data migration options for container workloads on Red Hat OpenShift clusters with Astra Control Center (ACC). Specifically, customers can use ACC to
- move some selected workloads or all workloads from their on-premises data centers to the cloud
- clone their apps to the cloud either for testing purposes or move from the data center to the cloud

Data Migration

To migrate application from one environment to another, you can use one of the following features of ACC:
- replication
- backup and restore
- clone

Refer to the data protection section for the replication and backup and restore options. Refer here for additional details about cloning.

Astra Replication feature is only supported with Trident Container Storage Interface (CSI). However, replication is not supported by nas-economy & san-economy drivers.

Performing data replication using ACC

NetApp Hybrid Multicloud solutions for Red Hat OpenShift Container workloads

Overview

NetApp is seeing a significant increase in customers modernizing their legacy enterprise applications and building new applications using containers and orchestration platforms built around Kubernetes. Red Hat OpenShift Container Platform is one example that we see adopted by many of our customers.

As more and more customers begin adopting containers within their enterprises, NetApp is perfectly positioned to help serve the persistent storage needs of their stateful applications and classic data management needs such as data protection, data security, and data migration. However, these needs are met using different strategies, tools, and methods.

NetApp ONTAP based storage options listed below, deliver security, data protection, reliability, and flexibility for containers and Kubernetes deployments.
• Self-managed storage in on-premises:
  ◦ NetApp Fabric Attached Storage (FAS), NetApp All Flash FAS Arrays (AFF), NetApp All SAN Array (ASA) and ONTAP Select

• Provider-managed storage in on-premises:
  ◦ NetApp Keystone provides Storage as a Service (STaaS)

• Self-managed storage in the cloud:
  ◦ NetApp Cloud Volumes ONTAP(CVO) provide self managed storage in the hyperscalers

• Provider-managed storage in the cloud:
  ◦ Cloud Volumes Service for Google Cloud (CVS), Azure NetApp Files (ANF), Amazon FSx for NetApp ONTAP offer fully managed storage in the hyperscalers

### ONTAP feature highlights

<table>
<thead>
<tr>
<th>Storage Administration</th>
<th>Performance &amp; Scalability</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Multi-tenancy</td>
<td>• ONTAP CLI &amp; API</td>
</tr>
<tr>
<td>• FlexVol &amp; FlexGroup</td>
<td>• System Manager &amp; BlueXP</td>
</tr>
<tr>
<td>• LUN</td>
<td></td>
</tr>
<tr>
<td>• Quotas</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Availability &amp; Resilience</th>
<th>Access Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Multi-AZ HA deployment</td>
<td>• NFS –v3, v4, v4.1, v4.2</td>
</tr>
<tr>
<td>(MetroCluster)</td>
<td>• iSCSI</td>
</tr>
<tr>
<td>• SnapShot &amp; SnapRestore</td>
<td>• SMB – v2, v3</td>
</tr>
<tr>
<td>• SnapMirror</td>
<td>• Multi-protocol access</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Storage Efficiency</th>
<th>Security &amp; Compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Deduplication &amp; Compression</td>
<td>• Fpolicy &amp; Vscan</td>
</tr>
<tr>
<td>• Compaction</td>
<td>• LDAP &amp; Kerberos</td>
</tr>
<tr>
<td></td>
<td>• Data Tiering (Fabric Pool)</td>
</tr>
<tr>
<td></td>
<td>• Certificate based authentication</td>
</tr>
</tbody>
</table>

**NetApp BlueXP** enables you to manage all of your storage and data assets from a single control plane/interface.

You can use BlueXP to create and administer cloud storage (for example, Cloud Volumes ONTAP and Azure NetApp Files), to move, protect, and analyze data, and to control many on-prem and edge storage devices.

**NetApp Astra Trident** is a CSI Compliant Storage Orchestrator that enable quick and easy consumption of persistent storage backed by a variety of the above-mentioned NetApp storage options. It is an open-source software maintained and supported by NetApp.
Business critical container workloads need more than just persistent volumes. Their data management requirements require protection and migration of the application kubernetes objects as well.

Application data includes kubernetes objects in addition to the user data: Some examples are as follows:
- kubernetes objects such as pods specs, PVCs, deployments, services
- custom config objects such as config maps and secrets
- persistent data such as Snapshot copies, backups, clones
- custom resources such as CRs and CRDs

NetApp Astra Control, available as both fully-managed and self-managed software, provides orchestration for robust application data management. Refer to the Astra documentation for additional details on the Astra family of products.

This reference documentation provides validation of migration and protection of container-based applications, deployed on RedHat OpenShift container platform, using NetApp Astra Control Center. In addition, the solution provides high-level details for the deployment and the use of Red Hat Advanced Cluster Management (ACM) for managing the container platforms. The document also highlights the details for the integration of NetApp storage with Red Hat OpenShift container platforms using Astra Trident CSI provisioner. Astra Control Center is deployed on the hub cluster and is used to manage the container applications and their persistent storage lifecycle. Finally, it provides a solution for replication and failover and fail-back for container workloads on managed Red Hat OpenShift clusters in AWS (ROSA) using Amazon FSx for NetApp ONTAP (FSxN) as persistent storage.

NetApp Solution with Managed Red Hat OpenShift Container platform workloads on AWS

Customers may be "born in the cloud" or may be at a point in their modernization journey when they are ready to move some select workloads or all workloads from their data centers to the cloud. They may choose to use provider-managed OpenShift containers and provider-managed NetApp storage in the cloud for running their workloads. They should plan and deploy the Managed Red Hat OpenShift container clusters (ROSA) in
the cloud for a successful production-ready environment for their container workloads. When they are in AWS cloud, they could also deploy FSx for NetApp ONTAP for the storage needs.

FSx for NetApp ONTAP delivers data protection, reliability, and flexibility for container deployments in AWS. Astra Trident serves as the dynamic storage provisioner to consume the persistent FSxN storage for customers’ stateful applications.

As ROSA can be deployed in HA mode with control plane nodes spread across multiple availability zones, FSx ONTAP can also be provisioned with Multi-AZ option which provides high availability and protect against AZ failures.

There are no data transfer charges when accessing an Amazon FSx file system from the file system’s preferred Availability Zone (AZ). For more info on pricing, refer [here](#).

Data protection and migration solution for OpenShift Container workloads

![Diagram showing ROSA clusters and FSxN setup](image)

#### Deploy and configure the Managed Red Hat OpenShift Container platform on AWS

This section describes a high-level workflow of setting up the Managed Red Hat OpenShift clusters on AWS (ROSA). It shows the use of Managed FSx for NetApp ONTAP (FSxN) as the storage backend by Astra Trident to provide persistent volumes. Details are provided about the deployment of FSxN on AWS using BlueXP. Also, details are provided about the use of BlueXP and OpenShift GitOps (Argo CD) to perform data protection and migration activities for the stateful applications on ROSA clusters.

Here is a diagram that depicts the ROSA clusters deployed on AWS and using FSxN as the backend storage.
This solution was verified by using two ROSA clusters in two VPCs in AWS. Each ROSA cluster was integrated with FSxN using Astra Trident. There are several ways of deploying ROSA clusters and FSxN in AWS. This high-level description of the setup provides documentation links for the specific method that was used. You can refer to the other methods in the relevant links provided in the resources section.

The setup process can be broken down into the following steps:

Install ROSA clusters

- Create two VPCs and set up VPC peering connectivity between the VPCs.
- Refer here for instructions to install ROSA clusters.

Install FSxN

- Install FSxN on the VPCs from BlueXP. Refer here for BlueXP account creation and to get started. Refer here for installing FSxN. Refer here for creating a connector in AWS to manage the FSxN.
- Deploy FSxN using AWS. Refer here for deployment using AWS console.
Install Trident on ROSA clusters (using Helm chart)

- Use Helm chart to install Trident on ROSA clusters.
  url for the Helm chart: https://netapp.github.io/trident-helm-chart

Integration of FSxN with Astra Trident for ROSA clusters

OpenShift GitOps can be utilized to deploy Astra Trident CSI to all managed clusters as they get registered to ArgoCD using ApplicationSet.

```yaml
apiVersion: argoproj.io/v1alpha1
kind: ApplicationSet
metadata:
  name: trident-operator
spec:
generators:
- clusters: {}
  # selector:
  # matchLabels:
  # tridentversion: '23.04.0'
template:
  metadata:
    name: '{(nameNormalized)}-trident'
spec:
  destination:
    namespace: trident
  server: '{(server)}'
source:
  repoURL: 'https://netapp.github.io/trident-helm-chart'
targetRevision: 23.04.0
  chart: trident-operator
  project: default
  syncPolicy:
  syncOptions:
    - CreateNamespace=true
```
Create backend and storage classes using Trident (for FSxN)

- Refer here for details about creating backend and storage class.
- Make the storage class created for FsxN with Trident CSI as default from OpenShift Console. See screenshot below:

Deploy an application using OpenShift GitOps (Argo CD)

- Install OpenShift GitOps operator on the cluster. Refer to instructions here.
- SetUp a new Argo CD instance for the cluster. Refer to instructions here.

Open the console of Argo CD and deploy an app.
As an example, you can deploy a Jenkins App using Argo CD with a Helm Chart.
When creating the application, the following details were provided:
Project: default
cluster: https://kubernetes.default.svc
Namespace: Jenkins
The url for the Helm Chart: https://charts.bitnami.com/bitnami
Helm Parameters:
global.storageClass: fsxn-nas

Data protection

This page shows the data protection options for Managed Red Hat OpenShift on AWS (ROSA) clusters using Astra Control Service. Astra Control Service (ACS) provides an easy-to-use graphical user-interface with which you can add clusters, define applications running on them, and perform application aware data management activities. ACS functions can also be accessed using an API that allows for automation of workflows.

Powering Astra Control (ACS or ACC) is NetApp Astra Trident. Astra Trident integrates several types of Kubernetes clusters such as Red Hat OpenShift, EKS, AKS, SUSE Rancher, Anthos etc., with various flavors of NetApp ONTAP storage such as FAS/AFF, ONTAP Select, CVO, Google Cloud Volumes Service, Azure
NetApp Files and Amazon FSx for NetApp ONTAP.

This section provides details for the following data protection options using ACS:

- A video showing Backup and Restore of a ROSA application running in one region and restoring to another region.
- A video showing Snapshot and Restore of a ROSA application.
- Step-by-step details of installing a ROSA cluster, Amazon FSx for NetApp ONTAP, using NetApp Astra Trident to integrate with storage backend, installing a postgresql application on ROSA cluster, using ACS to create a snapshot of the application and restoring the application from it.
- A blog showing step-by-step details of creating and restoring from a snapshot for a mysql application on a ROSA cluster with FSx for ONTAP using ACS.

Backup/Restore from Backup

The following video shows the backup of a ROSA application running in one region and restoring to another region.

FSx NetApp ONTAP for Red Hat OpenShift Service on AWS

Snapshot/Restore from snapshot

The following video shows taking a snapshot of a ROSA application and restoring from the snapshot after.

Snapshot/Restore for Applications on Red Hat OpenShift Service on AWS (ROSA)clusters with Amazon FSx for NetApp ONTAP storage

Blog

- Using Astra Control Service for data management of apps on ROSA clusters with Amazon FSx storage

Step-by-Step Details to create snapshot and restore from it

Prerequisite setup

- AWS account
- Red Hat OpenShift account
- IAM user with appropriate permissions to create and access ROSA cluster
- AWS CLI
- ROSA CLI
- OpenShift CLI(oc)
- VPC with subnets and appropriate gateways and routes
- ROSA Cluster installed into the VPC
- Amazon FSx for NetApp ONTAP created in the same VPC
- Access to the ROSA cluster from OpenShift Hybrid Cloud Console

Next Steps

1. Create an admin user and login to the cluster.
2. Create a kubeconfig file for the cluster.
3. Install Astra Trident on the cluster.
4. Create a backend, storage class and snapshot class configuration using the Trident CSI provisioner.
5. Deploy a postgresql application on the cluster.
6. Create a database and add a record.
7. Add the cluster into ACS.
8. Define the application in ACS.
9. Create a snapshot using ACS.
10. Delete the database in the postgresql application.
11. Restore from a snapshot using ACS.
12. Verify your app has been restored form the snapshot.

1. **Create an admin user and login to the cluster**

Access the ROSA cluster by creating an admin user with the following command: (You need to create an admin user only if you did not create one at the time of installation)

```
rosa create admin --cluster=<cluster-name>
```

The command will provide an output that will look like the following. Login to the cluster using the `oc login` command provided in the output.

```
W: It is recommended to add an identity provider to login to this cluster. See 'rosa create idp --help' for more information.
I: Admin account has been added to cluster 'my-rosa-cluster'. It may take up to a minute for the account to become active.
I: To login, run the following command:
    --username cluster-admin 
    --password FWGYL-ZmkJI-00000-00000
```

You can also login to the cluster using a token. If you already created an admin-user at the time of cluster creation, you can login to the cluster from the Red Hat OpenShift Hybrid Cloud console with the admin-user credentials. Then by clicking on the top right corner where it displays the name of the logged in user, you can obtain the `oc login` command (token login) for the command line.

2. **Create a kubeconfig file for the cluster**

Follow the procedures [here](#) to create a kubeconfig file for the ROSA cluster. This kubeconfig file will be used later when you add the cluster into ACS.
3. Install Astra Trident on the cluster

Install Astra Trident (latest version) on the ROSA cluster. To do this, you can follow any one of the procedures given here. To install Trident using helm from the console of the cluster, first create a project called Trident.

Then from the Developer view, create a Helm chart repository. For the URL field use 'https://netapp.github.io/trident-helm-chart'. Then create a helm release for Trident operator.
Create Helm Chart Repository

Add helm chart repository.

Configure via:  ○ Form view  ○ YAML view

Scope type
○ Namespaced scoped (ProjectHelmChartRepository)
  Add Helm Chart Repository in the selected namespace.
○ Cluster scoped (HelmChartRepository)
  Add Helm Chart Repository at the cluster level and in all namespaces.

Name *
trident
A unique name for the Helm Chart repository.

Display name
Astra Trident
A display name for the Helm Chart repository.

Description
NetApp Astra Trident
A description for the Helm Chart repository.

☐ Disable usage of the repo in the developer catalog.

URL *
https://netapp.github.io/trident-helm-chart
Verify all trident pods are running by going back to the Administrator view on the console and selecting pods in the trident project.
4. Create a backend, storage class and snapshot class configuration using the Trident CSI provisioner

Use the yaml files shown below to create a trident backend object, storage class object and the Volumesnapshot object. Be sure to provide the credentials to your Amazon FSx for NetApp ONTAP file system you created, the management LIF and the vserver name of your file system in the configuration yaml for the backend. To get those details, go to the AWS console for Amazon FSx and select the file system in the configuration yamlml for the backend. To get those details, go to the AWS console for Amazon FSx and select the file system, navigate to the Administration tab. Also, click on update to set the password for the fsxadmin user.

You can use the command line to create the objects or create them with the yaml files from the hybrid cloud console.
Trident Backend Configuration

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-nas-secret
type: Opaque
stringData:
  username: fsxadmin
  password: <password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: ontap-nas
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: <management lif>
  backendName: ontap-nas
  svm: fsx
  credentials:
    name: backend-tbc-ontap-nas-secret
```

Storage Class
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nas
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
allowVolumeExpansion: true

snapshot class

apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: trident-snapshotclass
driver: csi.trident.netapp.io
deletionPolicy: Delete

Verify that the backend, storage class and the trident-snapshotclass objects are created by issuing the commands shown below.

```
[ec2-user@ip-10-49-11-132 storage]$ kubectl get tbc -n trident
NAME   BACKEND NAME   BACKEND UID
ontap-nas   ontap-nas   8a5e4583-2dac-46bb-b01e-fa7c3816f121
```

```
[ec2-user@ip-10-49-11-132 storage]$ kubectl get sc
NAME    PROVISIONER   RECLAIM_POLICY   VOLUME_BINDING_MODE   ALLOW_VOLUME_EXPANSION   AGE
gp2     kubernetes.io/aws-ebs   Delete   WaitForFirstConsumer   true   3h23m
gp2-csi  ebs.csi.aws.com   Delete   WaitForFirstConsumer   true   3h19m
gp3-csi  ebs.csi.aws.com   Delete   WaitForFirstConsumer   true   3h19m
ontap-nas   csi.trident.netapp.io   Delete   Immediate   true   141m
```

At this time, an important modification you need to make is to set ontap-nas as the default storage class instead of gp3 so that the postgresql app you deploy later can use the default storage class. In the Openshift console of your cluster, under Storage select StorageClasses. Edit the annotation of the current default class to be false and add the annotation storageclass.kubernetes.io/is-default-class set to true for the ontap-nas storage class.
5. Deploy a postgresql application on the cluster

You can deploy the application from the command line as follows:

```
helm install postgresql bitnami/postgresql -n postgresql --create-namespace
```
If you do not see the application pods running, then there might be an error caused due to security context constraints.

Fix the error by editing the `runAsUser` and `fsGroup` fields in `statefulset.apps/postgresql` object with the uid that is in the output of the `oc get project` command as shown below.

Postgresql app should be running and using persistent volumes backed by Amazon FSx for NetApp ONTAP storage.
6. Create a database and add a record

```
[ec2-user@ip-10-49-11-132 astra]$ export POSTGRES_PASSWORD=$(kubectl get secret --namespace postgresql postgresql -o jsonpath="{.data.postgres-password}") | base64 -d
> --command -- psql -h postgres -U postgres -d postgres -p 5432
Warning: would violate PodSecurity "restricted": allowPrivilegeEscalation=false (container "postgresql-client" must set securityContext.capabilities.drop="["ALL"]", runAsNonRoot=true (pod or container "postgresql-client" must set securityContext.runAsNonRoot=true), seccompProfile (pod or container "postgresql-client" must set securityContext.seccompProfile.type to "RuntimeDefault" or "Localhost")
If you don't see a command prompt, try pressing enter.

postgres=# CREATE DATABASE erp;
CREATE DATABASE
postgres=# c erp
You are now connected to database "erp" as user "postgres".

erp=# CREATE TABLE PERSONS(ID INT PRIMARY KEY NOT NULL, FIRSTNAME TEXT NOT NULL, LASTNAME TEXT NOT NULL);
CREATE TABLE

erp=# INSERT INTO PERSONS VALUES(1, 'John', 'Doe');
INSERT 0 1

erp=# \dt
List of relations

<table>
<thead>
<tr>
<th>Schema</th>
<th>Name</th>
<th>Type</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>public</td>
<td>persons</td>
<td>table</td>
<td>postgres</td>
</tr>
</tbody>
</table>

(1 row)

erp=# SELECT * FROM persons;
<table>
<thead>
<tr>
<th>id</th>
<th>firstname</th>
<th>lastname</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John</td>
<td>Doe</td>
</tr>
</tbody>
</table>
(1 row)
```

7. Add the cluster into ACS

Log in to ACS. Select cluster and click on Add. Select other and upload or paste the kubeconfig file.
Click **Next** and select **ontap-nas** as the default storage class for ACS. Click **Next**, review the details and **Add** the cluster.

8. Define the application in ACS

Define the postgresql application in ACS. From the landing page, select **Applications**, **Define** and fill in the appropriate details. Click **Next** a couple of times, Review the details and click **Define**. The application gets
9. Create a snapshot using ACS

There are many ways to create a snapshot in ACS. You can select the application and create a snapshot from the page that shows the details of the application. You can click on Create snapshot to create an on-demand snapshot or configure a protection policy.

Create an on-demand snapshot by simply clicking on Create snapshot, providing a name, reviewing the details, and clicking on Snapshot. The snapshot state changes to Healthy after the operation is completed.
10. Delete the database in the postgresql application

Log back into postgresql, list the available databases, delete the one you created previously and list again to ensure that the database has been deleted.

```
postgres=# \l
  Name | Owner  | Encoding | Locale Provider | Collate       | Ctype       | ICU Locale | ICU Rules | Access privilege
    ---+--------+----------+-----------------+--------------+-------------+------------+-----------+-------------------
     erp | postgres| UTF8     | libc            | en_US.UTF-8  | en_US.UTF-8 |             |           |                   
 postgres | postgres| UTF8     | libc            | en_US.UTF-8  | en_US.UTF-8 |             |           |                   
template0 | postgres| UTF8     | libc            | en_US.UTF-8  | en_US.UTF-8 |             |           |                   
 template1 | postgres| UTF8     | libc            | en_US.UTF-8  | en_US.UTF-8 |             |           |                   
        (4 rows)
```

```
postgres=# DROP DATABASE erp;
DROP DATABASE
postgres=# \l
  Name | Owner  | Encoding | Locale Provider | Collate       | Ctype       | ICU Locale | ICU Rules | Access privilege
    ---+--------+----------+-----------------+--------------+-------------+------------+-----------+-------------------
 postgres | postgres| UTF8     | libc            | en_US.UTF-8  | en_US.UTF-8 |             |           |                   
template0 | postgres| UTF8     | libc            | en_US.UTF-8  | en_US.UTF-8 |             |           |                   
 template1 | postgres| UTF8     | libc            | en_US.UTF-8  | en_US.UTF-8 |             |           |                   
        (3 rows)
```

11. Restore from a snapshot using ACS

To restore the application from a snapshot, go to ACS UI landing page, select the application and select
Restore. You need to pick a snapshot or a backup from which to restore. (Typically, you would have multiple
created based on a policy that you have configured). Make appropriate choices in the next couple of screens
and then click on **Restore**. The application status moves from Restoring to Available after it has been restored
from the snapshot.
12. Verify your app has been restored from the snapshot

Login to the postgresql client and you should now see the table and the record in the table that you previously had. That's it. Just by clicking a button, your application has been restored to a previous state. That is how easy we make it for our customers with Astra Control.

```
$ oc get postgresql -l app=postgres
$ oc get statefulset -l app=postgres
```

Data migration

This page shows the data migration options for container workloads on Managed Red Hat OpenShift clusters using FSx for NetApp ONTAP for persistent storage.
Data Migration

Red Hat OpenShift service on AWS as well as FSx for NetApp ONTAP (FSxN) are part of their service portfolio by AWS. FSxN is available on Single AZ or Multi-AZ options. Multi-Az option provides data protection from availability zone failure. FSxN can be integrated with Astra Trident to provide persistent storage for applications on ROSA clusters.

Integration of FSxN with Trident using Helm chart

ROSA Cluster Integration with Amazon FSx for ONTAP

The migration of container applications involves:

• Persistent volumes: this can be accomplished using BlueXP. Another option is to use Astra Control Center to handle container application migrations from on-premises to the cloud environment. Automation can be used for the same purpose.

• Application metadata: this can be accomplished using OpenShift GitOps (Argo CD).

Failover and Fail-back of applications on ROSA cluster using FSxN for persistent storage

The following video is a demonstration of application failover and fail-back scenarios using BlueXP and Argo CD.

Failover and Fail-back of applications on ROSA cluster

Data protection and migration solution for OpenShift Container workloads
NetApp Solutions for Virtualization

Introduction to automation for ONTAP and vSphere

VMware automation

Automation has been an integral part of managing VMware environments since the first days of VMware ESX. The ability to deploy infrastructure as code and extend practices to private cloud operations helps to alleviate concerns surrounding scale, flexibility, self-provisioning, and efficiency.

Automation can be organized into the following categories:

- Virtual infrastructure deployment
- Guest machine operations
- Cloud operations

There are many options available to administrators with respect to automating their infrastructure. Whether through using native vSphere features such as Host Profiles or Customization Specifications for virtual machines to available APIs on the VMware software components, operating systems, and NetApp storage systems; there is significant documentation and guidance available.

Data ONTAP 8.0.1 and later supports certain VMware vSphere APIs for Array Integration (VAAI) features when the ESX host is running ESX 4.1 or later. VAAI is a set of APIs that enable communication between VMware vSphere ESXi hosts and storage devices. These features help offload operations from the ESX host to the storage system and increase network throughput. The ESX host enables the features automatically in the correct environment. You can determine the extent to which your system is using VAAI features by checking the statistics contained in the VAAI counters.

The most common starting point for automating the deployment of a VMware environment is provisioning block or file-based datastores. It is important to map out the requirements of the actual tasks prior to developing the corresponding automation.

For more information concerning the automation of VMware environments, see the following resources:

- The Ansible Galaxy Community for VMware. A collection of Ansible resources for VMware.
- VMware {code} Resources. Resources needed to design solutions for the software-defined data center, including forums, design standards, sample code, and developer tools.

vSphere traditional block storage provisioning with ONTAP

VMware vSphere supports the following VMFS datastore options with ONTAP SAN protocol support indicated.

<table>
<thead>
<tr>
<th>VMFS datastore options</th>
<th>ONTAP SAN protocol support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fibre Channel (FC)</td>
<td>yes</td>
</tr>
<tr>
<td>Fibre Channel over Ethernet (FCoE)</td>
<td>yes</td>
</tr>
</tbody>
</table>
### VMFS datastore options

<table>
<thead>
<tr>
<th></th>
<th>ONTAP SAN protocol support</th>
</tr>
</thead>
<tbody>
<tr>
<td>iSCSI</td>
<td>yes</td>
</tr>
<tr>
<td>iSCSI Extensions for RDMA (iSER)</td>
<td>no</td>
</tr>
<tr>
<td>NVMe over Fabric with FC (NVMe/FC)</td>
<td>yes</td>
</tr>
<tr>
<td>NVMe over Fabric with RDMA over Converged Ethernet (NVMe/RoCE)</td>
<td>no</td>
</tr>
</tbody>
</table>

If iSER or NVMe/RoCE VMFS is required, check SANtricity-based storage systems.

### vSphere VMFS datastore - Fibre Channel storage backend with ONTAP

#### About this task

This section covers the creation of a VMFS datastore with ONTAP Fibre Channel (FC) storage.

For automated provisioning, use one of these scripts: [PowerShell], Ansible Playbook, or [Terraform].

#### What you need

- The basic skills necessary to manage a vSphere environment and ONTAP
- An ONTAP storage system (FAS/AFF/CVO/ONTAP Select/ASA) running ONTAP 9.8 or later
- ONTAP credentials (SVM name, userID, and password)
- ONTAP WWPN of host, target, and SVM and LUN information
- The completed FC configuration worksheet
- vCenter Server credentials
- vSphere host(s) information
  - vSphere 7.0 or later
- Fabric switch(es)
  - With connected ONTAP FC data ports and vSphere hosts
  - With the N_port ID virtualization (NPIV) feature enabled
  - Create a single initiator single target zone.
    - Create one zone for each initiator (single initiator zone).
    - For each zone, include a target that is the ONTAP FC logical interface (WWPN) for the SVMs. There should be at least two logical interfaces per node per SVM. Do not use the WWPN of the physical ports.
- An ONTAP Tool for VMware vSphere deployed, configured, and ready to consume.

#### Provisioning a VMFS datastore

To provision a VMFS datastore, complete the following steps:

1. Check compatibility with the [Interoperability Matrix Tool (IMT)]
2. Verify that the FCP Configuration is supported.
ONTAP tasks

1. Verify that you have an ONTAP license for FCP.
   a. Use the system license show command to check that FCP is listed.
   b. Use license add -license-code <license code> to add the license.

2. Make sure that the FCP protocol is enabled on the SVM.
   a. Verify the FCP on an existing SVM.
   b. Configure the FCP on an existing SVM.
   c. Create a new SVM with the FCP.

3. Make sure that FCP logical interfaces are available on an SVM.
   a. Use Network Interface show to verify the FCP adapter.
   b. When an SVM is created with the GUI, logical interfaces are a part of that process.
   c. To rename network interfaces, use Network Interface modify.

4. Create and Map a LUN. Skip this step if you are using ONTAP tools for VMware vSphere.

VMware vSphere tasks

1. Verify that HBA drivers are installed. VMware supported HBAs have drivers deployed out of the box and should be visible in the Storage Adapter Information.

2. Provision a VMFS datastore with ONTAP Tools.

vSphere VMFS Datastore - Fibre Channel over Ethernet storage protocol with ONTAP

About this task

This section covers the creation of a VMFS datastore with the Fibre Channel over Ethernet (FCoE) transport protocol to ONTAP storage.

For automated provisioning, use one of these scripts: [PowerShell], Ansible Playbook, or [Terraform].

What you need

• The basic skills necessary to manage a vSphere environment and ONTAP
• An ONTAP storage system (FAS/AFF/CVO/ONTAP Select) running ONTAP 9.8 or later
• ONTAP credentials (SVM name, userID, and password)
• A supported FCoE combination
• A completed configuration worksheet
• vCenter Server credentials
• vSphere host(s) information
  ◦ vSphere 7.0 or later
• Fabric switch(es)
  ◦ With either ONTAP FC data ports or vSphere hosts connected
  ◦ With the N_port ID virtualization (NPIV) feature enabled
Create a single initiator single target zone.
- FC/FCoE zoning configured
- Network switch(es)
  - FCoE support
  - DCB support
  - Jumbo frames for FCoE
- ONTAP Tool for VMware vSphere deployed, configured, and ready to consume

Provision a VMFS datastore
- Check compatibility with the Interoperability Matrix Tool (IMT).
- Verify that the FCoE configuration is supported.

ONTAP tasks
1. Verify the ONTAP license for FCP.
   a. Use the `system license show` command to verify that the FCP is listed.
   b. Use `license add -license-code <license code>` to add a license.
2. Verify that the FCP protocol is enabled on the SVM.
   a. Verify the FCP on an existing SVM.
   b. Configure the FCP on an existing SVM.
   c. Create a new SVM with the FCP.
3. Verify that FCP logical interfaces are available on the SVM.
   a. Use `Network Interface show` to verify the FCP adapter.
   b. When the SVM is created with the GUI, logical interfaces are a part of that process.
   c. To rename the network interface, use `Network Interface modify`.
4. Create and map a LUN; skip this step if you are using ONTAP tools for VMware vSphere.

VMware vSphere tasks
1. Verify that HBA drivers are installed. VMware-supported HBAs have drivers deployed out of the box and should be visible in the storage adapter information.
2. Provision a VMFS datastore with ONTAP Tools.

vSphere VMFS Datastore - iSCSI Storage backend with ONTAP

About this task
This section covers the creation of a VMFS datastore with ONTAP iSCSI storage.

For automated provisioning, use one of these scripts: [PowerShell], Ansible Playbook, or [Terraform].

What you need
- The basic skills necessary to manage a vSphere environment and ONTAP.
• An ONTAP storage system (FAS/AFF/CVO/ONTAP Select/ASA) running ONTAP 9.8 or later
• ONTAP credentials (SVM name, userID, and password)
• ONTAP network port, SVM, and LUN information for iSCSI
• A completed iSCSI configuration worksheet
• vCenter Server credentials
• vSphere host(s) information
  ◦ vSphere 7.0 or later
• iSCSI VMKernel adapter IP information
• Network switch(es)
  ◦ With ONTAP system network data ports and connected vSphere hosts
  ◦ VLAN(s) configured for iSCSI
  ◦ (Optional) link aggregation configured for ONTAP network data ports
• ONTAP Tool for VMware vSphere deployed, configured, and ready to consume

Steps
1. Check compatibility with the Interoperability Matrix Tool (IMT).
2. Verify that the iSCSI configuration is supported.
3. Complete the following ONTAP and vSphere tasks.

ONTAP tasks
1. Verify the ONTAP license for iSCSI.
   a. Use the system license show command to check if iSCSI is listed.
   b. Use license add -license-code <license code> to add the license.
2. Verify that the iSCSI protocol is enabled on the SVM.
3. Verify that iSCSI network logical interfaces are available on the SVM.
   • When an SVM is created using the GUI, iSCSI network interfaces are also created.
4. Use the Network interface command to view or make changes to the network interface.
   • Two iSCSI network interfaces per node are recommended.
5. Create an iSCSI network interface. You can use the default-data-blocks service policy.
6. Verify that the data-iscsi service is included in the service policy. You can use network interface service-policy show to verify.
7. Verify that jumbo frames are enabled.
8. Create and map the LUN. Skip this step if you are using ONTAP tools for VMware vSphere. Repeat this step for each LUN.
VMware vSphere tasks

1. Verify that at least one NIC is available for the iSCSI VLAN. Two NICs are preferred for better performance and fault tolerance.

2. Identify the number of physical NICs available on the vSphere host.

3. Configure the iSCSI initiator. A typical use case is a software iSCSI initiator.

4. Verify that the TCPIP stack for iSCSI is available.

5. Verify that iSCSI portgroups are available.
   ◦ We typically use a single virtual switch with multiple uplink ports.
   ◦ Use 1:1 adapter mapping.

6. Verify that iSCSI VMKernel adapters are enabled to match the number of NICs and that IPs are assigned.

7. Bind the iSCSI software adapter to the iSCSI VMKernel adapter(s).

8. Provision the VMFS datastore with ONTAP Tools. Repeat this step for all datastores.

9. Verify hardware acceleration support.

What's next?

After these the tasks are completed, the VMFS datastore is ready to consume for provisioning virtual machines.

Ansible Playbook

```yaml
## Disclaimer: Sample script for reference purpose only.

- hosts: '{{ vsphere_host }}'
  name: Play for vSphere iSCSI Configuration
  connection: local
  gather_facts: false
  tasks:
    # Generate Session ID for vCenter
    - name: Generate a Session ID for vCenter
      uri:
        url: "https://{{ vcenter_hostname }}/rest/com/vmware/cis/session"
        validate_certs: false
        method: POST
        user: "{{ vcenter_username }}"
        password: "{{ vcenter_password }}"
        force_basic_auth: yes
        return_content: yes
        register: vclogin

    # Generate Session ID for ONTAP tools with vCenter
    - name: Generate a Session ID for ONTAP tools with vCenter
      uri:
        url: "https://{{ ontap_tools_ip }}:8143/api/rest/2.0/security/user/login"
```
validate_certs: false
method: POST
return_content: yes
body_format: json
body:
  vcenterUserName: "{{ vcenter_username }}"
  vcenterPassword: "{{ vcenter_password }}"
register: login

# Get existing registered ONTAP Cluster info with ONTAP tools
- name: Get ONTAP Cluster info from ONTAP tools
  uri:
    url: "https://{{ ontap_tools_ip }}:8143/api/rest/2.0/storage/clusters"
    validate_certs: false
    method: Get
    return_content: yes
    headers:
      vmware-api-session-id: "{{ login.json.vmwareApiSessionId }}"
    register: clusterinfo

- name: Get ONTAP Cluster ID
  set_fact:
    ontap_cluster_id: "{{ clusterinfo.json | json_query(clusteridquery) }}"
  vars:
    clusteridquery: "records[?ipAddress == '{{ netapp_hostname }}' && type=='Cluster'].id | [0]"

- name: Get ONTAP SVM ID
  set_fact:
    ontap_svm_id: "{{ clusterinfo.json | json_query(svmidquery) }}"
  vars:
    svmidquery: "records[?ipAddress == '{{ netapp_hostname }}' && type=='SVM' && name == '{{ svm_name }}'].id | [0]"

- name: Get Aggregate detail
  uri:
    url: "https://{{ ontap_tools_ip }}:8143/api/rest/2.0/storage/clusters/{{ ontap_svm_id }}/aggregates"
    validate_certs: false
    method: GET
    return_content: yes
    headers:
      vmware-api-session-id: "{{ login.json.vmwareApiSessionId }}"
      cluster-id: "{{ ontap_svm_id }}"
when: ontap_svm_id != ''
register: aggrinfo

- name: Select Aggregate with max free capacity
  set_fact:
    aggr_name: "{{ aggrinfo.json | json_query(aggrquery) }}"
  vars:
    aggrquery: "max_by(records, &freeCapacity).name"

- name: Convert datastore size in MB
  set_fact:
    datastoreSizeInMB: "{{ iscsi_datastore_size | human_to_bytes/1024/1024 | int }}"

- name: Get vSphere Cluster Info
  uri:
    url: "https://{{ vcenter_hostname }}/api/vcenter/cluster?names="{{ vsphere_cluster }}"
    validate_certs: false
    method: GET
    return_content: yes
    body_format: json
    headers:
      vmware-api-session-id: "{{ vclogin.json.value }}"
  when: vsphere_cluster != ''
  register: vcenterclusterid

- name: Create iSCSI VMFS-6 Datastore with ONTAP tools
  uri:
    url: "https://{{ ontap_tools_ip }}:8143/api/rest/3.0/admin/datastore"
    validate_certs: false
    method: POST
    return_content: yes
    status_code: [200]
    body_format: json
    body:
      traditionalDatastoreRequest:
        name: "{{ iscsi_datastore_name }}"
        datastoreType: VMFS
        protocol: ISCSI
        spaceReserve: Thin
        clusterID: "{{ ontap_cluster_id }}"
        svmID: "{{ ontap_svm_id }}"
        targetMoref: ClusterComputeResource:{{ vcenterclusterid.json[0].cluster }}"
vSphere VMFS Datastore - NVMe/FC with ONTAP

About this task

This section covers the creation of a VMFS datastore with ONTAP storage using NVMe/FC.

For automated provisioning, use one of these scripts: [PowerShell], Ansible Playbook, or [Terraform].

What you need

- Basic skills needed to manage a vSphere environment and ONTAP.
- Basic understanding of NVMe/FC.
- An ONTAP Storage System (FAS/AFF/CVO/ONTAP Select/ASA) running ONTAP 9.8 or later
- ONTAP credentials (SVM name, userID, and password)
- ONTAP WWPN for host, target, and SVMs and LUN information
- A completed FC configuration worksheet
- vCenter Server
- vSphere host(s) information (vSphere 7.0 or later)
- Fabric switch(es)
  - With ONTAP FC data ports and vSphere hosts connected.
  - With the N_port ID virtualization (NPIV) feature enabled.
  - Create a single initiator target zone.
  - Create one zone for each initiator (single initiator zone).
  - For each zone, include a target that is the ONTAP FC logical interface (WWPN) for the SVMs. There should be at least two logical interfaces per node per SVM. DO not use the WWPN of physical ports.

Provision VMFS datastore

1. Check compatibility with the Interoperability Matrix Tool (IMT).
2. Verify that the NVMe/FC configuration is supported.
ONTAP tasks

1. **Verify the ONTAP license for FCP.**
   Use the `system license show` command and check if NVMe_oF is listed.
   Use `license add -license-code <license code>` to add a license.

2. **Verify that NVMe protocol is enabled on the SVM.**
   a. Configure SVMs for NVMe.

3. **Verify that NVMe/FC Logical Interfaces are available on the SVMs.**
   a. Use `Network Interface show` to verify the FCP adapter.
   b. When an SVM is created with the GUI, logical interfaces are as part of that process.
   c. To rename the network interface, use the command `Network Interface modify`.

4. **Create NVMe namespace and subsystem**

VMware vSphere Tasks

1. **Verify that HBA drivers are installed.** VMware supported HBAs have the drivers deployed out of the box and should be visible at Storage Adapter Information
2. **Perform vSphere Host NVMe driver installation and validation tasks**
3. **Create VMFS Datastore**

vSphere traditional file storage provisioning with ONTAP

VMware vSphere supports following NFS protocols, both of which support ONTAP.

- NFS Version 3
- NFS Version 4.1

If you need help selecting the correct NFS version for vSphere, check this comparison of NFS client versions.

Reference

vSphere datastore and protocol features: NFS

vSphere NFS datastore - Version 3 with ONTAP

About this task

Creation of NFS version 3 datastore with ONTAP NAS storage.

For automated provisioning, use one of these scripts: [PowerShell], Ansible Playbook, or [Terraform].

What you need

- The basic skill necessary to manage a vSphere environment and ONTAP.
- An ONTAP storage system (FAS/AFF/CVO/ONTAP Select/Cloud Volume Service/Azure NetApp Files) running ONTAP 9.8 or later
- ONTAP credentials (SVM name, userID, password)
- ONTAP network port, SVM, and LUN information for NFS
A completed NFS configuration worksheet

vCenter Server credentials

vSphere host(s) information for vSphere 7.0 or later

NFS VMKernel adapter IP information

Network switch(es)
  • with ONTAP system network data ports and connected vSphere hosts
  • VLAN(s) configured for NFS
  • (Optional) link aggregation configured for ONTAP network data ports

ONTAP Tool for VMware vSphere deployed, configured, and ready to consume

Steps

• Check compatibility with the Interoperability Matrix Tool (IMT)
  • Verify that the NFS configuration is supported.
• Complete the following ONTAP and vSphere tasks.

ONTAP tasks

1. Verify the ONTAP license for NFS.
   a. Use the `system license show` command and check that NFS is listed.
   b. Use `license add -license-code <license code>` to add a license.

2. Follow the NFS configuration workflow.

VMware vSphere Tasks

Follow the workflow for NFS client configuration for vSphere.

Reference

vSphere datastore and protocol features: NFS

What's next?

After these tasks are completed, the NFS datastore is ready to consume for provisioning virtual machines.

vSphere NFS Datastore - Version 4.1 with ONTAP

About this task

This section describes the creation of an NFS version 4.1 datastore with ONTAP NAS storage.

For automated provisioning, use one of these scripts: [PowerShell], Ansible Playbook, or [Terraform].

What you need

• The basic skills necessary to manage a vSphere environment and ONTAP
• ONTAP Storage System (FAS/AFF/CVO/ONTAP Select/Cloud Volume Service/Azure NetApp Files)
running ONTAP 9.8 or later

• ONTAP credentials (SVM name, userID, password)
• ONTAP network port, SVM, and LUN information for NFS
• A completed NFS configuration worksheet
• vCenter Server credentials
• vSphere host(s) information vSphere 7.0 or later
• NFS VMKernel adapter IP information
• Network switch(es)
  ◦ with ONTAP system network data ports, vSphere hosts, and connected
  ◦ VLAN(s) configured for NFS
  ◦ (Optional) link aggregation configured for ONTAP network data ports
• ONTAP Tools for VMware vSphere deployed, configured, and ready to consume

Steps

• Check compatibility with the Interoperability Matrix Tool (IMT).
  ◦ Verify that the NFS configuration is supported.
• Complete the ONTAP and vSphere Tasks provided below.

ONTAP tasks

1. Verify ONTAP license for NFS
   a. Use the `system license show` command to check whether NFS is listed.
   b. Use `license add -license-code <license code>` to add a license.

2. Follow the NFS configuration workflow

VMware vSphere tasks

Follow the NFS Client Configuration for vSphere workflow.

What's next?

After these tasks are completed, the NFS datastore is ready to consume for provisioning virtual machines.

NetApp Hybrid Multicloud with VMware Solutions

VMware Hybrid Multicloud Use Cases

Use Cases for NetApp Hybrid Multicloud with VMware

An overview of the use cases of importance to IT organization when planning hybrid-cloud or cloud-first deployments.
Popular Use Cases

Use cases include:

- Disaster recovery,
- Hosting workloads during data center maintenance, * quick burst in which additional resources are required beyond what’s provisioned in the local data center,
- VMware site expansion,
- Fast migration to the cloud,
- Dev/test, and
- Modernization of apps leveraging cloud supplemental technologies.

Throughout this documentation, cloud workload references will be detailed using the VMware use-cases. These use-cases are:

- Protect (includes both Disaster Recovery and Backup / Restore)
- Migrate
- Extend

Inside the IT Journey

Most organizations are on a journey to transformation and modernization. As part of this process, companies are trying use their existing VMware investments while leveraging cloud benefits and exploring ways to make the migration process as seamless as possible. This approach would make their modernization efforts very easy because the data is already in the cloud.

The easiest answer to this scenario is VMware offerings in each hyperscaler. Like NetApp® Cloud Volumes, VMware provides a way to move or extend on-premises VMware environments to any cloud, allowing you to retain existing on-premises assets, skills, and tools while running workloads natively in the cloud. This reduces risk because there will be no service breaks or a need for IP changes and provides the IT team the ability to operate the way they do on-premises using existing skills and tools. This can lead to accelerated cloud migrations and a much smoother transition to a hybrid Multicloud architecture.

Understanding the Importance of Supplemental NFS Storage Options

While VMware in any cloud delivers unique hybrid capabilities to every customer, limited supplemental NFS storage options have restricted its usefulness for organizations with storage-heavy workloads. Because storage is directly tied to hosts, the only way to scale storage is to add more hosts—and that can increase costs by 35–40 percent or more for storage intensive workloads. These workloads just need additional storage, not additional horsepower. But that means paying for additional hosts.

Let’s consider this scenario:

A customer requires just five hosts for CPU and memory, but has a lot of storage needs, and needs 12 hosts to meet the storage requirement. This requirement ends up really tipping the financial scale by having to buy the additional horsepower, when they only need to increment the storage.

When you’re planning cloud adoption and migrations, it’s always important to evaluate the best approach and take the easiest path that reduces total investments. The most common and easiest approach for any application migration is rehosting (also known as lift and shift) where there is no virtual machine (VM) or data conversion. Using NetApp Cloud Volumes with VMware software-defined data center (SDDC), while complementing vSAN, provides an easy lift-and-shift option.
Virtual Desktops

Virtual Desktop Services (VDS)

TR-4861: Hybrid Cloud VDI with Virtual Desktop Service

Suresh Thoppay, NetApp

The NetApp Virtual Desktop Service (VDS) orchestrates Remote Desktop Services (RDS) in major public clouds as well as on private clouds. VDS supports Windows Virtual Desktop (WVD) on Microsoft Azure. VDS automates many tasks that must be performed after deployment of WVD or RDS, including setting up SMB file shares (for user profiles, shared data, and the user home drive), enabling Windows features, application and agent installation, firewall, and policies, and so on.

Users consume VDS for dedicated desktops, shared desktops, and remote applications. VDS provides scripted events for automating application management for desktops and reduces the number of images to manage.

VDS provides a single management portal for handling deployments across public and private cloud environments.

Customer Value

The remote workforce explosion of 2020 has changed requirements for business continuity. IT departments are faced with new challenges to rapidly provision virtual desktops and thus require provisioning agility, remote management, and the TCO advantages of a hybrid cloud that makes it easy to provision on-premises and cloud resources. They need a hybrid-cloud solution that:

- Addresses the post-COVID workspace reality to enable flexible work models with global dynamics
- Enables shift work by simplifying and accelerating the deployment of work environments for all employees, from task workers to power users
- Mobilizes your workforce by providing rich, secure VDI resources regardless of the physical location
- Simplifies hybrid-cloud deployment
- Automates and simplifies risk reduction management

Use Cases

Hybrid VDI with NetApp VDS allows service providers and enterprise virtual desktop administrators to easily expand resources to other cloud environment without affecting their users. Having on-premises resources provides better control of resources and offers wide selection of choices (compute, GPU, storage, and network) to meet demand.

This solution applies to the following use cases:

- Bursting into the cloud for surges in demand for remote desktops and applications
- Reducing TCO for long running remote desktops and applications by hosting them on-premises with flash storage and GPU resources
- Ease of management of remote desktops and applications across cloud environments
• Experience remote desktops and applications by using a software-as-a-service model with on-premises resources

Target Audience

The target audience for the solution includes the following groups:

• EUC/VDI architects who wants to understand the requirements for a hybrid VDS
• NetApp partners who would like to assist customers with their remote desktop and application needs
• Existing NetApp HCI customers who want to address remote desktop and application demands

NetApp Virtual Desktop Service Overview

NetApp offers many cloud services, including the rapid provisioning of virtual desktop with WVD or remote applications and rapid integration with Azure NetApp Files.

Traditionally, it takes weeks to provision and deliver remote desktop services to customers. Apart from provisioning, it can be difficult to manage applications, user profiles, shared data, and group policy objects to enforce policies. Firewall rules can increase complexity and require a separate skillset and tools.

With Microsoft Azure Windows Virtual Desktop service, Microsoft takes care of maintenance for Remote Desktop Services components, allowing customers to focus on provisioning workspaces in the cloud. Customers must provision and manage the complete stack which requires special skills to manage VDI environments.

With NetApp VDS, customers can rapidly deploy virtual desktops without worrying about where to install the architecture components like brokers, gateways, agents, and so on. Customers who require complete control of their environment can work with a professional services team to achieve their goals. Customers consume VDS as a service and thus can focus on their key business challenges.

NetApp VDS is a software-as-a-service offering for centrally managing multiple deployments across AWS, Azure, GCP, or private cloud environments. Microsoft Windows Virtual Desktop is available only on Microsoft Azure. NetApp VDS orchestrates Microsoft Remote Desktop Services in other environments.

Microsoft offers multisession on Windows 10 exclusively for Windows Virtual Desktop environments on Azure. Authentication and identity are handled by the virtual desktop technology; WVD requires Azure Active Directory synced (with AD Connect) to Active Directory and session VMs joined to Active Directory. RDS requires Active Directory for user identity and authentication and VM domain join and management.

A sample deployment topology is shown in the following figure.
Each deployment is associated with an active directory domain and provides clients with an access entry point for workspaces and applications. A service provider or enterprise that has multiple active directory domains typically has more deployments. A single Active Directory domain that spans multiple regions typically has a single deployment with multiple sites.

For WVD in Azure, Microsoft provides a platform-as-a-service that is consumed by NetApp VDS. For other environments, NetApp VDS orchestrates the deployment and configuration of Microsoft Remote Desktop Services. NetApp VDS supports both WVD Classic and WVD ARM and can also be used to upgrade existing versions.

Each deployment has its own platform services, which consists of Cloud Workspace Manager (REST API endpoint), an HTML 5 Gateway (connect to VMs from a VDS management portal), RDS Gateways (Access point for clients), and a Domain Controller. The following figure depicts the VDS Control Plane architecture for RDS implementation.
For RDS implementations, NetApp VDS can be readily accessed from Windows and browsers using client software that can be customized to include customer logo and images. Based on user credentials, it provides user access to approved workspaces and applications. There is no need to configure the gateway details.

The following figure shows the NetApp VDS client.
In the Azure WVD implementation, Microsoft handles the access entry point for the clients and can be consumed by a Microsoft WVD client available natively for various OSs. It can also be accessed from a web-based portal. The configuration of client software must be handled by the Group Policy Object (GPO) or in other ways preferred by customers.

The following figure depicts the VDS Control Plane architecture for Azure WVD implementations.
In addition to the deployment and configuration of required components, NetApp VDS also handles user management, application management, resource scaling, and optimization.

NetApp VDS can create users or grant existing user accounts access to cloud workspace or application services. The portal can also be used for password resets and the delegation of administrating a subset of components. Helpdesk administrators or Level-3 technicians can shadow user sessions for troubleshooting or connect to servers from within the portal.

NetApp VDS can use image templates that you create, or it can use existing ones from the marketplace for cloud-based provisioning. To reduce the number of images to manage, you can use a base image, and any additional applications that you require can be provisioned using the provided framework to include any command-line tools like Chocolatey, MSIX app attach, PowerShell, and so on. Even custom scripts can be used as part of machine lifecycle events.

**NetApp HCI Overview**

NetApp HCI is a hybrid cloud infrastructure that consists of a mix of storage nodes and compute nodes. It is available as either a two-rack unit or single-rack unit, depending on the model. The installation and configuration required to deploy VMs are automated with the NetApp Deployment Engine (NDE). Compute clusters are managed with VMware vCenter, and storage clusters are managed with the vCenter Plug-in deployed with NDE. A management VM called the mNode is deployed as part of the NDE.

NetApp HCI handles the following functions:

- Version upgrades
- Pushing events to vCenter
- vCenter Plug-In management
- A VPN tunnel for support
- The NetApp Active IQ collector
- The extension of NetApp Cloud Services to on the premises, enabling a hybrid cloud infrastructure. The following figure depicts HCI components.
Storage Nodes

Storage nodes are available as either a half-width or full-width rack unit. A minimum of four storage nodes is required at first, and a cluster can expand to up to 40 nodes. A storage cluster can be shared across multiple compute clusters. All the storage nodes contain a cache controller to improve write performance. A single node provides either 50K or 100K IOPS at a 4K block size.

NetApp HCI storage nodes run NetApp Element software, which provides minimum, maximum, and burst QoS limits. The storage cluster supports a mix of storage nodes, although one storage node cannot exceed one-third of total capacity.

Compute Nodes

NetApp supports its storage connected to any compute servers listed in the VMware Compatibility Guide.

Compute nodes are available in half-width, full-width, and two rack-unit sizes. The NetApp HCI H410C and H610C are based on scalable Intel Skylake processors. The H615C is based on second-generation scalable Intel Cascade Lake processors. There are two compute models that contain GPUs: the H610C contains two NVIDIA M10 cards and the H615C contains three NVIDIA T4 cards.

The NVIDIA T4 has 40 RT cores that provide the computation power needed to deliver real-time ray tracing. The same server model used by designers and engineers can now also be used by artists to create photorealistic imagery that features light bouncing off surfaces just as it would in real life. This RTX-capable GPU produces real-time ray tracing performance of up to five Giga Rays per second. The NVIDIA T4, when combined with Quadro Virtual Data Center Workstation (Quadro vDWS) software, enables artists to create photorealistic designs with accurate shadows, reflections, and refractions on any device from any location.

Tensor cores enable you to run deep learning inferencing workloads. When running these workloads, an NVIDIA T4 powered with Quadro vDWS can perform up to 25 times faster than a VM driven by a CPU-only server. A NetApp H615C with three NVIDIA T4 cards in one rack unit is an ideal solution for graphics and compute-intensive workloads.

The following figure lists NVIDIA GPU cards and compares their features.
The M10 GPU remains the best TCO solution for knowledge-worker use cases. However, the T4 makes a great alternative when IT wants to standardize on a GPU that can be used across multiple use cases, such as virtual workstations, graphics performance, real-time interactive rendering, and inferencing. With the T4, IT can take advantage of the same GPU resources to run mixed workloads—for example, running VDI during the day and repurposing the resources to run compute workloads at night.

The H610C compute node is two rack units in size; the H615C is one rack unit in size and consumes less power. The H615C supports H.264 and H.265 (High Efficiency Video Coding [HEVC]) 4:4:4 encoding and decoding. It also supports the increasingly mainstream VP9 decoder; even the WebM container package served by YouTube uses the VP9 codec for video.

The number of nodes in a compute cluster is dictated by VMware; currently, it is 96 with VMware vSphere 7.0 Update 1. Mixing different models of compute nodes in a cluster is supported when Enhanced vMotion Compatibility (EVC) is enabled.

**NVIDIA Licensing**

When using an H610C or H615C, the license for the GPU must be procured from NVIDIA partners that are authorized to resell the licenses. You can find NVIDIA partners with the partner locator. Search for competencies such as virtual GPU (vGPU) or Tesla.

NVIDIA vGPU software is available in four editions:

- NVIDIA GRID Virtual PC (GRID vPC)
- NVIDIA GRID Virtual Applications (GRID vApps)
- NVIDIA Quadro Virtual Data Center Workstation (Quadro vDWS)
- NVIDIA Virtual ComputeServer (vComputeServer)
GRID Virtual PC

This product is ideal for users who want a virtual desktop that provides a great user experience for Microsoft Windows applications, browsers, high-definition video, and multi-monitor support. The NVIDIA GRID Virtual PC delivers a native experience in a virtual environment, allowing you to run all your PC applications at full performance.

GRID Virtual Applications

GRID vApps are for organizations deploying a Remote Desktop Session Host (RDSH) or other app-streaming or session-based solutions. Designed to deliver Microsoft Windows applications at full performance, Windows Server-hosted RDSH desktops are also supported by GRID vApps.

Quadro Virtual Data Center Workstation

This edition is ideal for mainstream and high-end designers who use powerful 3D content creation applications like Dassault CATIA, SOLIDWORKS, 3Dexcite, Siemens NX, PTC Creo, Schlumberger Petrel, or Autodesk Maya. NVIDIA Quadro vDWS allows users to access their professional graphics applications with full features and performance anywhere on any device.

NVIDIA Virtual ComputeServer

Many organizations run compute-intensive server workloads such as artificial intelligence (AI), deep learning (DL), and data science. For these use cases, NVIDIA vComputeServer software virtualizes the NVIDIA GPU, which accelerates compute-intensive server workloads with features such as error correction code, page retirement, peer-to-peer over NVLink, and multi-vGPU.

A Quadro vDWS license enables you to use GRID vPC and NVIDIA vComputeServer.

Deployment

NetApp VDS can be deployed to Microsoft Azure using a setup app available based on the required codebase. The current release is available here and the preview release of the upcoming product is available here.

See this video for deployment instructions.
Hybrid Cloud Environment

NetApp Virtual Desktop Service can be extended to on-premises when connectivity exists between on-premises resources and cloud resources. Enterprises can establish the link to Microsoft Azure using Express Route or a site-to-site IPsec VPN connection. You can also create links to other clouds in a similar way either using a dedicated link or with an IPsec VPN tunnel.

For the solution validation, we used the environment depicted in the following figure.
On-premises, we had multiple VLANs for management, remote-desktop-session hosts, and so on. They were on the 172.21.146-150.0/24 subnet and routed to the corporate network using the Microsoft Remote Routing Access Service. We also performed the following tasks:

1. We noted the public IP of the Microsoft Routing and Remote Access Server (RRAS; identified with IPchicken.com).
2. We created a Virtual Network Gateway resource (route-based VPN) on Azure Subscription.
3. We created the connection providing the local network gateway address for the public IP of the Microsoft RRAS server.
4. We completed VPN configuration on RRAS to create a virtual interface using pre-shared authentication that was provided while creating the VPN gateway. If configured correctly, the VPN should be in the connected state. Instead of Microsoft RRAS, you can also use pfSense or other relevant tools to create the site-to-site IPsec VPN tunnel. Since it is route-based, the tunnel redirects traffic based on the specific subnets configured.

Microsoft Azure Active Directory provides identity authentication based on oAuth. Enterprise client authentications typically require NTLM or Kerberos-based authentication. Microsoft Azure Active Directory Domain Services perform password hash sync between Azure Active Directory and on-prem domain controllers using ADConnect.

For this Hybrid VDS solution validation, we initially deployed to Microsoft Azure and added an additional site with vSphere. The advantage with this approach is that platform services were deployed to Microsoft Azure and were then readily backed up using the portal. Services can then be easily accessed from anywhere, even if the site-site VPN link is down.

To add another site, we used a tool called DCConfig. The shortcut to that application is available on the desktop of the cloud workspace manager (CWMgr) VM. After this application is launched, navigate to the DataCenter Sites tab, add the new datacenter site, and fill in the required info as shown below. The URL points to the vCenter IP. Make sure that the CWMgr VM can communicate with vCenter before adding the
Make sure that vSphere PowerCLI 5.1 on CloudWorkspace manager is installed to enable communication with VMware vSphere environment.

The following figure depicts on-premises datacenter site configuration.

Note that there are filtering options available for compute resource based on the specific cluster, host name, or free RAM space. Filtering options for storage resource includes the minimum free space on datastores or the maximum VMs per datastore. Datastores can be excluded using regular expressions. Click Save button to save the configuration.

To validate the configuration, click the Test button or click Load Hypervisor and check any dropdown under the vSphere section. It should be populated with appropriate values. It is a best practice to keep the primary hypervisor set to yes for the default provisioning site.

The VM templates created on VMware vSphere are consumed as provisioning collections on VDS. Provisioning collections come in two forms: shared and VDI. The shared provisioning collection type is used for remote desktop services for which a single resource policy is applied to all servers. The VDI type is used for WVD instances for which the resource policy is individually assigned. The servers in a provisioning collection can be assigned one of the following three roles:
• **TSDATA.** Combination of Terminal Services and Data server role.

• **TS.** Terminal Services (Session Host).

• **DATA.** File Server or Database Server. When you define the server role, you must pick the VM template and storage (datastore). The datastore chosen can be restricted to a specific datastore or you can use the least-used option in which the datastore is chosen based on data usage.

Each deployment has VM resource defaults for the cloud resource allocation based on Active Users, Fixed, Server Load, or User Count.

**Single server load test with Login VSI**

The NetApp Virtual Desktop Service uses the Microsoft Remote Desktop Protocol to access virtual desktop sessions and applications, and the Login VSI tool determines the maximum number of users that can be hosted on a specific server model. Login VSI simulates user login at specific intervals and performs user operations like opening documents, reading and composing mails, working with Excel and PowerPoint, printing documents, compressing files, and taking random breaks. It then measures response times. User response time is low when server utilization is low and increases when more user sessions are added. Login VSI determines the baseline based on initial user login sessions and it reports the maximum user session when the user response exceeds 2 seconds from the baseline.

NetApp Virtual Desktop Service utilizes Microsoft Remote Desktop Protocol to access the Virtual Desktop session and Applications. To determine the maximum number of users that can be hosted on a specific server model, we used the Login VSI tool. Login VSI simulates user login at specific intervals and performs user operations like opening documents, reading and composing mails, working with Excel and PowerPoint, printing documents, compressing files, taking random breaks, and so on. It also measures response times. User response time is low when server utilization is low and increases when more user sessions are added. Login VSI determines the baseline based on initial user login sessions and it reports maximum user sessions when the user response exceeds 2 sec from the baseline.

The following table contains the hardware used for this validation.

<table>
<thead>
<tr>
<th>Model</th>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetApp HCI H610C</td>
<td>4</td>
<td>Three in a cluster for launchers, AD, DHCP, and so on. One server for load testing.</td>
</tr>
<tr>
<td>NetApp HCI H615C</td>
<td>1</td>
<td>2x24C Intel Xeon Gold 6282 @2.1GHz. 1.5TB RAM.</td>
</tr>
</tbody>
</table>

The following table contains the software used for this validation.

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetApp VDS 5.4</td>
<td>Orchestration</td>
</tr>
<tr>
<td>VM Template Windows 2019</td>
<td>Server OS for RDSH</td>
</tr>
<tr>
<td>1809</td>
<td></td>
</tr>
<tr>
<td>Login VSI</td>
<td>4.1.32.1</td>
</tr>
<tr>
<td>VMware vSphere 6.7 Update 3</td>
<td>Hypervisor</td>
</tr>
</tbody>
</table>
The Login VSI test results are as follows:

<table>
<thead>
<tr>
<th>Model</th>
<th>VM configuration</th>
<th>Login VSI baseline</th>
<th>Login VSI Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>H610C</td>
<td>8 vCPU, 48GB RAM, 75GB disk, 8Q vGPU profile</td>
<td>799</td>
<td>178</td>
</tr>
<tr>
<td>H615C</td>
<td>12 vCPU, 128GB RAM, 75GB disk</td>
<td>763</td>
<td>272</td>
</tr>
</tbody>
</table>

Considering sub-NUMA boundaries and hyperthreading, the eight VMs chosen for VM testing and configuration depended on the cores available on the host.

We used 10 launcher VMs on the H610C, which used the RDP protocol to connect to the user session. The following figure depicts the Login VSI connection information.

The following figure displays the Login VSI response time versus the active sessions for the H610C.
The following figure displays the Login VSI response time versus active sessions for the H615C.

The performance metrics from Cloud Insights during H615C Login VSI testing for the vSphere host and VMs are shown in the following figure.
Management Portal

NetApp VDS Cloud Workspace Management Suite portal is available here and the upcoming version is available here.

The portal allows centralized management for various VDS deployments including one that has sites defined for on-premises, administrative users, the application catalog, and scripted events. The portal is also used by administrative users for the manual provisioning of applications if required and to connect to any machines for troubleshooting.

Service providers can use this portal to add their own channel partners and allow them to manage their own clients.

User Management

NetApp VDS uses Azure Active Directory for identity authentication and Azure Active Directory Domain Services for NTLM/Kerberos authentication. The ADConnect tool can be used to sync an on-prem Active Directory domain with Azure Active Directory.

New users can be added from the portal, or you can enable cloud workspace for existing users. Permissions for workspaces and application services can be controlled by individual users or by groups. From the management portal, administrative users can be defined to control permissions for the portal, workspaces, and so on.

The following figure depicts user management in NetApp VDS.
Each workspace resides in its own Active Directory organization unit (OU) under the Cloud Workspace OU as shown in the following figure.

For more info, see this video on user permissions and user management in NetApp VDS.

When an Active Directory group is defined as a CRAUserGroup using an API call for the datacenter, all the users in that group are imported into the CloudWorkspace for management using the UI. As the cloud workspace is enabled for the user, VDS creates user home folders, settings permissions, user properties updates, and so on.
If VDI User Enabled is checked, VDS creates a single-session RDS machine dedicated to that user. It prompts for the template and the datastore to provision.

Workspace Management

A workspace consists of a desktop environment; this can be shared remote desktop sessions hosted on-premises or on any supported cloud environment. With Microsoft Azure, the desktop environment can be persistent with Windows Virtual Desktops. Each workspace is associated with a specific organization or client. Options available when creating a new workspace can be seen in the following figure.
Each workspace is associated with a specific deployment.

Workspaces contain associated apps and app services, shared data folders, servers, and a WVD instance. Each workspace can control security options like enforcing password complexity, multifactor authentication, file audits, and so on.

Workspaces can control the workload schedule to power on extra servers, limit the number of users per server, or set the schedule for the resources available for a given period (always on/off). Resources can also be configured to wake up on demand.

The workspace can override the deployment VM resource defaults if required. For WVD, WVD host pools (which contains session hosts and app groups) and WVD workspaces can also be managed from the cloud workspace management suite portal. For more info on the WVD host pool, see this video.

**Application Management**

Task workers can quickly launch an application from the list of applications made available to them. App services publish applications from the Remote Desktop Services session hosts. With WVD, App Groups provide similar functionality from multi-session Windows 10 host pools.

For office workers to power users, the applications that they require can be provisioned manually using a service board, or they can be auto-provisioned using the scripted events feature in NetApp VDS.

For more information, see the NetApp Application Entitlement page.
ONTAP features for Virtual Desktop Service

The following ONTAP features make it an attractive choice for use with a virtual desktop service.

- **Scale-out filesystem.** ONTAP FlexGroup volumes can grow to more than 20PB in size and can contain more than 400 billion files within a single namespace. The cluster can contain up to 24 storage nodes, each with a flexible number of network interface cards depending on the model used.

  User’s virtual desktops, home folders, user profile containers, shared data, and so on can grow based on demand with no concern for filesystem limitations.

- **File system analytics.** You can use the XCP tool to gain insights into shared data. With ONTAP 9.8+ and ActiveIQ Unified Manager, you can easily query and retrieve file metadata information and identify cold data.

- **Cloud tiering.** You can migrate cold data to an object store in the cloud or to any S3-compatible storage in your datacenter.

- **File versions.** Users can recover files protected by NetApp ONTAP Snapshot copies. ONTAP Snapshot copies are very space efficient because they only record changed blocks.

- **Global namespace.** ONTAP FlexCache technology allows remote caching of file storage making it easier to manage shared data across locations containing ONTAP storage systems.

- **Secure multi-tenancy support.** A single physical storage cluster can be presented as multiple virtual storage arrays each with its own volumes, storage protocols, logical network interfaces, identity and authentication domain, management users, and so on. Therefore, you can share the storage array across multiple business units or environments, such as test, development, and production.

  To guarantee performance, you can use adaptive QoS to set performance levels based on used or allocated space, and you can control storage capacity by using quotas.

- **VMware integration.** ONTAP tools for VMware vSphere provide a vCenter plug-in to provision datastores, implement vSphere host best practices, and monitor ONTAP resources.

  ONTAP supports vStorage APIs for Array Integration (VAAI) for offloading SCSI/file operations to the storage array. ONTAP also supports vStorage APIs for Storage Awareness (VASA) and Virtual Volumes support for both block and file protocols.

  The Snapcenter Plug-in for VMware vSphere provides an easy way to back up and restore virtual machines using the Snapshot feature on a storage array.

  ActiveIQ Unified Manager provides end-to-end storage network visibility in a vSphere environment. Administrators can easily identify any latency issues that might occur on virtual desktop environments hosted on ONTAP.

- **Security compliance.** With ActiveIQ Unified Manager, you can monitor multiple ONTAP systems with alerts for any policy violations.

- **Multi-protocol support.** ONTAP supports block (iSCSI, FC, FCoE, and NVMe/FC), file (NFSv3, NFSv4.1, SMB2.x, and SMB3.x), and object (S3) storage protocols.

- **Automation support.** ONTAP provides REST API, Ansible, and PowerShell modules to automate tasks with the VDS Management Portal.
Data Management

As a part of deployment, you can choose the file-services method to host the user profile, shared data, and the home drive folder. The available options are File Server, Azure Files, or Azure NetApp Files. However, after deployment, you can modify this choice with the Command Center tool to point to any SMB share. There are various advantages to hosting with NetApp ONTAP. To learn how to change the SMB share, see Change Data Layer.

Global File Cache

When users are spread across multiple sites within a global namespace, Global File Cache can help reduce latency for frequently accessed data. Global File Cache deployment can be automated using a provisioning collection and scripted events. Global File Cache handles the read and write caches locally and maintains file locks across locations. Global File Cache can work with any SMB file servers, including Azure NetApp Files.

Global File Cache requires the following:

- Management server (License Management Server)
- Core
• Edge with enough disk capacity to cache the data

To download the software and to calculate the disk cache capacity for Edge, see the GFC documentation.

For our validation, we deployed the core and management resources on the same VM at Azure and edge resources on NetApp HCI. Please note that the core is where high-volume data access is required and the edge is a subset of the core. After the software is installed, you must activate the license activated before use. To do so, complete the following steps:

1. Under the License Configuration section, use the link Click Here to complete the license activation. Then register the core.

2. Provide the service account to be used for the Global File Cache. For the required permissions for this account, see the GFC documentation.
3. Add a new backend file server and provide the file server name or IP.

4. On the edge, the cache drive must have the drive letter D. If it does not, use `diskpart.exe` to select the volume and change drive letter. Register with the license server as edge.
If core auto-configuration is enabled, core information is retrieved from the license management server automatically.

From any client machine, the administrators that used to access the share on the file server can access it with GFC edge using UNC Path `\\<edge server name>\FASTDATA\<core server name>\<backend file server name>\<share name>`. Administrators can include this path in user logonscript or GPO for users drive mapping at the edge location.

To provide transparent access for users across the globe, an administrator can setup the Microsoft Distributed
Filesystem (DFS) with links pointing to file server shares and to edge locations.

When users log in with Active Directory credentials based on the subnets associated with the site, the appropriate link is utilized by the DFS client to access the data.

File icons change depending on whether a file is cached; files that are not cached have a grey X on the lower left corner of the icon. After a user in an edge location accesses a file, that file is cached, and the icon changes.
When a file is open and another user is trying to open the same file from an edge location, the user is prompted with the following selection:

![File In Use](image)

If the user selects the option to receive a notification when the original copy is available, the user is notified as follows:

![File Now Available](image)

For more information, see this video on Talon and Azure NetApp Files Deployment.

**SaaS Backup**

NetApp VDS provides data protection for Salesforce and Microsoft Office 365, including Exchange, SharePoint, and Microsoft OneDrive. The following figure shows how NetApp VDS provides SaaS Backup for these data services.
For a demonstration of Microsoft Office 365 data protection, see this video.

For a demonstration of Salesforce data protection, see this video.

Operation management

With NetApp VDS, administrators can delegate tasks to others. They can connect to deployed servers to troubleshoot, view logs, and run audit reports. While assisting customers, helpdesk or level-3 technicians can shadow user sessions, view process lists, and kill processes if required.

For information on VDS logfiles, see the Troubleshooting Failed VDA Actions page.

For more information on the required minimum permissions, see the VDA Components and Permissions page.

If you would like to manually clone a server, see the Cloning Virtual Machines page.
To automatically increase the VM disk size, see the Auto-Increase Disk Space Feature page.

To identify the gateway address to manually configure the client, see the End User Requirements page.

**Cloud Insights**

NetApp Cloud Insights is a web-based monitoring tool that gives you complete visibility into infrastructure and applications running on NetApp and other third-party infrastructure components. Cloud Insights supports both private cloud and public clouds for monitoring, troubleshooting, and optimizing resources.

Only the acquisition unit VM (can be Windows or Linux) must be installed on a private cloud to collect metrics from data collectors without the need for agents. Agent-based data collectors allow you to pull custom metrics from Windows Performance Monitor or any input agents that Telegraf supports.

The following figure depicts the Cloud Insights VDS dashboard.

For more info on NetApp Cloud Insights, see this video.

**Tools and Logs**

**DCConfig Tool**

The DCCconfig tool supports the following hypervisor options for adding a site:
Workspace-specific drive-letter mapping for shared data can be handled using GPO. Professional Services or the support team can use the advanced tab to customize settings like Active Directory OU names, the option to enable or disable deployment of FSLogix, various timeout values, and so on.
Command Center (Previously known as TestVdc Tools)

To launch Command Center and the required role, see the Command Center Overview.

You can perform the following operations:

• Change the SMB Path for a workspace.
• Change the site for provisioning collection.
Log Files
Check automation logs for more info.

**GPU considerations**

GPUs are typically used for graphic visualization (rendering) by performing repetitive arithmetic calculations. This repetitive compute capability is often used for AI and deep learning use cases.

For graphic intensive applications, Microsoft Azure offers the NV series based on the NVIDIA Tesla M60 card with one to four GPUs per VM. Each NVIDIA Tesla M60 card includes two Maxwell-based GPUs, each with 8GB of GDDR5 memory for a total of 16GB.

ℹ️ An NVIDIA license is included with the NV series.
With NetApp HCI, the H615C GPU contains three NVIDIA Tesla T4 cards. Each NVIDIA Tesla T4 card has a
Touring-based GPU with 16GB of GDDR6 memory. When used in a VMware vSphere environment, virtual machines are able to share the GPU, with each VM having dedicated frame buffer memory. Ray tracing is available with the GPUs on the NetApp HCI H615C to produce realistic images including light reflections. Please note that you need to have an NVIDIA license server with a license for GPU features.
To use the GPU, you must install the appropriate driver, which can be downloaded from the NVIDIA license.
portal. In an Azure environment, the NVIDIA driver is available as GPU driver extension. Next, the group policies in the following screenshot must be updated to use GPU hardware for remote desktop service sessions. You should prioritize H.264 graphics mode and enable encoder functionality.

Validate GPU performance monitoring with Task Manager or by using the nvidia-smi CLI when running WebGL samples. Make sure that GPU, memory, and encoder resources are being consumed.
To make sure that the virtual machine is deployed to the NetApp HCI H615C with Virtual Desktop Service, define a site with the vCenter cluster resource that has H615C hosts. The VM template must have the required vGPU profile attached.

For shared multi-session environments, consider allocating multiple homogenous vGPU profiles. However, for high end professional graphics application, it is better to have each VM dedicated to a user to keep VMs isolated.

The GPU processor can be controlled by a QoS policy, and each vGPU profile can have dedicated frame buffers. However, the encoder and decoder are shared for each card. The placement of a vGPU profile on a GPU card is controlled by the vSphere host GPU assignment policy, which can emphasize performance (spread VMs) or consolidation (group VMs).

Solutions for Industry

Graphics workstations are typically used in industries such as manufacturing, healthcare, energy, media and entertainment, education, architecture, and so on. Mobility is often limited for graphics-intensive applications.
To address the issue of mobility, Virtual Desktop Services provide a desktop environment for all types of workers, from task workers to expert users, using hardware resources in the cloud or with NetApp HCI, including options for flexible GPU configurations. VDS enables users to access their work environment from anywhere with laptops, tablets, and other mobile devices.

To run manufacturing workloads with software like ANSYS Fluent, ANSYS Mechanical, Autodesk AutoCAD, Autodesk Inventor, Autodesk 3ds Max, Dassault Systèmes SOLIDWORKS, Dassault Systèmes CATIA, PTC Creo, Siemens PLM NX, and so on, the GPUs available on various clouds (as of Jan 2021) are listed in the following table.

<table>
<thead>
<tr>
<th>GPU Model</th>
<th>Microsoft Azure</th>
<th>Google Compute (GCP)</th>
<th>Amazon Web Services (AWS)</th>
<th>On-Premises (NetApp HCI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA M60</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>NVIDIA T4</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>NVIDIA P100</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>NVIDIA P4</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Shared desktop sessions with other users and dedicated personal desktops are also available. Virtual desktops can have one to four GPUs or can utilize partial GPUs with NetApp HCI. The NVIDIA T4 is a versatile GPU card that can address the demands of a wide spectrum of user workloads.

Each GPU card on NetApp HCI H615C has 16GB of frame buffer memory and three cards per server. The number of users that can be hosted on single H615C server depends on the user workload.

<table>
<thead>
<tr>
<th>Users/Server</th>
<th>Light (4GB)</th>
<th>Medium (8GB)</th>
<th>Heavy (16GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>H615C</td>
<td>12</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

To determine the user type, run the GPU profiler tool while users are working with applications performing typical tasks. The GPU profiler captures memory demands, the number of displays, and the resolution that users require. You can then pick the vGPU profile that satisfies your requirements.

Virtual desktops with GPUs can support a display resolution of up to 8K, and the utility nView can split a single monitor into regions to work with different datasets.

With ONTAP file storage, you can realize the following benefits:

- A single namespace that can grow up to 20PB of storage with 400 billion of files, without much administrative input
- A namespace that can span the globe with a Global File Cache
- Secure multitenancy with managed NetApp storage
- The migration of cold data to object stores using NetApp FabricPool
- Quick file statistics with file system analytics
- Scaling a storage cluster up to 24 nodes increasing capacity and performance
- The ability to control storage space using quotas and guaranteed performance with QoS limits
- Securing data with encryption
- Meeting broad requirements for data protection and compliance
- Delivering flexible business continuity options
Conclusion

The NetApp Virtual Desktop Service provides an easy-to-consume virtual desktop and application environment with a sharp focus on business challenges. By extending VDS with the on-premises ONTAP environment, you can use powerful NetApp features in a VDS environment, including rapid clone, in-line deduplication, compaction, thin provisioning, and compression. These features save storage costs and improve performance with all-flash storage. With VMware vSphere hypervisor, which minimizes server-provisioning time by using Virtual Volumes and vSphere API for Array integration. Using the hybrid cloud, customers can pick the right environment for their demanding workloads and save money. The desktop session running on-premises can access cloud resources based on policy.

Where to Find Additional Information

To learn more about the information that is described in this document, review the following documents and/or websites:

- NetApp Cloud
- NetApp VDS Product Documentation
- Connect your on-premises network to Azure with VPN Gateway
- Azure Portal
- Microsoft Windows Virtual Desktop
- Azure NetApp Files Registration

VMware Horizon

NVA-1132-DESIGN: VMware end-user computing with NetApp HCI

Suresh Thoppay, NetApp

VMware end-user computing with NetApp HCI is a prevalidated, best-practice data center architecture for deploying virtual desktop workloads at an enterprise scale. This document describes the architectural design and best practices for deploying the solution at production scale in a reliable and risk-free manner.

NVA-1132-DESIGN: VMware end-user computing with NetApp HCI

NVA-1129-DESIGN: VMware end-user computing with NetApp HCI and NVIDIA GPUs

Suresh Thoppay, NetApp

VMware end-user computing with NetApp HCI is a prevalidated, best-practice data center architecture for deploying virtual desktop workloads at an enterprise scale. This document describes the architectural design and best practices for deploying the solution at production scale in a reliable and risk-free manner.

NVA-1129-DESIGN: VMware end-user computing with NetApp HCI and NVIDIA GPUs
VMware end-user Computing with NetApp HCI is a prevalidated, best-practice, data center architecture for deploying virtual desktop workloads at an enterprise scale. This document describes how to deploy the solution at production scale in a reliable and risk-free manner.

NetApp HCI for virtual desktop infrastructure with VMware Horizon 7 - Empower your power users with 3D Graphics

Suresh Thoppay, NetApp

TR-4792 provides guidance on using the NetApp H615C compute node for 3D graphics workloads in a VMware Horizon environment powered by NVIDIA graphics processing units (GPUs) and virtualization software. It also provides the results from the preliminary testing of SPECviewperf 13 for the H615C.

FlexPod desktop virtualization solutions

Learn more about FlexPod virtualization solutions by reviewing the FlexPod design guides

NetApp All-Flash SAN Array with VMware vSphere 8

Author: Josh Powell - NetApp Solutions Engineering

Solution Overview

Introduction

For nearly two decades, NetApp ONTAP software has established itself as a premier storage solution for VMware vSphere environments, continually introducing innovative features that simplify management and decrease costs. NetApp is an established leader in the development of NAS and unified storage platforms that offer a wide range of protocol and connectivity support. Alongside this market segment, there are many customers who prefer the simplicity and cost benefits of block-based SAN storage platforms that are focused on doing one job well. NetApp’s All-Flash SAN Array (ASA) delivers on that promise with simplicity at scale and with consistent management and automation features for all applications and cloud providers.

Purpose of This Document

In this document we will cover the unique value of using NetApp ASA storage systems with VMware vSphere and provide a technology overview of the NetApp All-Flash SAN Array. In addition, we will look at additional tools for simplifying storage provisioning, data protection, and monitoring of your VMware and ONTAP datacenter.
Technology Overview

This solution includes innovative technologies from VMware and NetApp.

VMware vSphere 8.0

VMware vSphere is a virtualization platform that transforms physical resources into pools of compute, network and storage which can be used to satisfy customers’ workload and application requirements. The main components of VMware vSphere include:

- **ESXi** - VMware’s hypervisor which enables the abstraction of compute processors, memory, network and other resources and makes them available to virtual machines and container workloads.

- **vCenter** - VMware vCenter is a centralized management platform for interacting with compute resources, networking and storage as part of a virtual infrastructure. vCenter plays a crucial role in simplifying the administration of virtualized infrastructure.

New Improvements in vSphere 8.0

vSphere 8.0 introduces some new improvements including, but not limited to:

- **Scalability** - vSphere 8.0 supports the latest Intel and AMD CPUs and has extended limits for vGPU devices, ESXi hosts, VMs per cluster, and VM DirectPath I/O devices.

- **Distributed Services Engine** - Network offloading with NSX to Data Processing Units (DPUs).

- **Enhanced Device Efficiency** - vSphere 8.0 boosts device management capabilities with features like device groups and Device Virtualization Extensions (DVX).

- **Improved Security** - The inclusion of an SSH timeout and TPM Provision Policy strengthens the security framework.

- **Integration with Hybrid Cloud Services** - This feature facilitates seamless transition between on-premises and cloud workloads.

- **Integrated Kubernetes Runtime** - With the inclusion of Tanzu, vSphere 8.0 simplifies container orchestration.

For more information refer to the blog, What’s New in vSphere 8?.

VMware Virtual Volumes (vVols)

vVols are a revolutionary new approach to storage management in vSphere clusters, providing simplified management and more granular control of storage resources. In a vVols datastore each virtual disk is a vVol and becomes a native LUN object on the storage system. The integration of the storage system and vSphere takes place through the **VMware API’s for Storage Awareness (VASA)** provider and allows the storage system to be aware of the VM data and manage it accordingly. Storage policies, defined in the vCenter Client are used to allocate and manage storage resources.

vVols are a simplified approach to storage management and are preferred in some use cases.

For more information on vVols see the vVols Getting Started Guide.
NVMe over Fabrics

With the release of vSphere 8.0, NVMe is now supported end-to-end with full support for vVols with NVMe-TCP and NVMe-FC.

For detailed information on using NVMe with vSphere refer to About VMware NVMe Storage in the vSphere Storage documentation.

NetApp ONTAP

NetApp ONTAP software has been a leading storage solution for VMware vSphere environments for almost two decades and continues to add innovative capabilities to simplify management while reducing costs. Using ONTAP together with vSphere is a great combination that lets you reduce host hardware and VMware software expenses. You can also protect your data at lower cost with consistent high performance while taking advantage of native storage efficiencies.

Base ONTAP Features

NetApp Snapshot copies: Snapshot copies of a VM or datastore, ensuring no performance impact upon the creation or utilization of a Snapshot. These replicas can serve as restoration points for VMs or as a simple data safeguard. These array-based snapshots are different than VMware (consistency) snapshots. The most straightforward method to generate an ONTAP Snapshot copy is through the SnapCenter Plug-In for VMware vSphere, backing up VMs and datastores.

- **Storage Efficiency** - ONTAP provides real-time and background deduplication and compression, zero-block deduplication, and data compaction.
- **Volume and LUN move** - Allows non-disruptive movement of volumes and LUNs supporting vSphere datastores and vVols within the ONTAP cluster to balance performance and capacity or support non-disruptive maintenance and upgrades.
- **Relocation of Volume and LUN** - ONTAP allows non-disruptive movement of volumes and LUNs that host vSphere datastores and vVols within the ONTAP cluster. This aids in balancing performance and capacity, and allows for non-disruptive upgrades.
- **Quality of Service** - QoS is a feature that enables the management of performance on an individual LUN, volume, or file. It can be used to limit an aggressive VM or to ensure that a critical VM receives sufficient performance resources.
- **Encryption** - NetApp Volume Encryption and NetApp Aggregate Encryption. These options provide a straightforward software-based approach to encrypting data at rest, ensuring its protection.
- **Fabric Pool** - This feature tiers less frequently accessed data to a separate object store, freeing up valuable flash storage. By operating at the block level, it efficiently identifies and tiers colder data, helping to optimize storage resources and reduce costs.
- **Automation** - Simplifies storage and data management tasks by utilizing ONTAP REST APIs for automation, and leveraging Ansible modules for seamless configuration management of ONTAP systems. Ansible modules offer a convenient solution for efficiently managing the configurations of ONTAP systems. The combination of these powerful tools enables the streamlining of workflows and enhancement of the overall management of storage infrastructure.

ONTAP Disaster Recovery Features

NetApp ONTAP provides robust disaster recovery solutions for VMware environments. These solutions leverage SnapMirror replication technologies between primary and secondary storage systems to allow failover.
and quick recovery in the case of failure.

**Storage Replication Adapter:**
The NetApp Storage Replication Adapter (SRA) is a software component that provides integration between NetApp storage systems and VMware Site Recovery Manager (SRM). It facilitates replication of virtual machine (VM) data across NetApp storage arrays, delivering robust data protection and disaster recovery capabilities. The SRA uses SnapMirror and SnapVault to achieve the replication of VM data across disparate storage systems or geographical locations.

The adapter provides asynchronous replication at the storage virtual machine (SVM) level using SnapMirror technology and extends support for both VMFS in SAN storage environments (iSCSI and FC) and NFS in NAS storage environments.

The NetApp SRA is installed as part of ONTAP Tools for VMware vSphere.

**SnapMirror Business Continuity:**
SnapMirror is a NetApp data replication technology that provides synchronous replication of data between storage systems. It allows for the creation of multiple copies of data at different locations, providing the ability to recover data in case of a disaster or data loss event. SnapMirror provides flexibility in terms of replication frequency and allows for the creation of point-in-time copies of data for backup and recovery purposes. SM-BC replicates data at the Consistency Group level.
NetApp MetroCluster:
NetApp MetroCluster is a high-availability and disaster recovery solution that provides synchronous data replication between two geographically dispersed NetApp storage systems. It is designed to ensure continuous data availability and protection in the event of a site-wide failure.

MetroCluster uses SyncMirror to synchronously replicate data just above the RAID level. SyncMirror is designed to efficiently transition between synchronous and asynchronous modes. This allows the primary storage cluster to continue operating in a non-replicated state in situations where the secondary site becomes temporarily inaccessible. SyncMirror will also replicate back to a RPO = 0 state when connectivity is restored.

MetroCluster can operate over IP based networks or using fibre channel.

For detailed information on MetroCluster architecture and configuration refer to the MetroCluster documentation site.
ONTAP One Licensing Model

ONTAP One is a comprehensive licensing model that provides access to all features of ONTAP without requiring additional licenses. This includes data protection, disaster recovery, high availability, cloud integration, storage efficiency, performance, and security. Customers with NetApp storage systems licensed with Flash, Core plus Data Protection, or Premium are entitled to ONTAP One licensing, ensuring they can maximize the use of their storage systems.

ONTAP One licensing includes all of the following features:

- **NVMeoF** – Enables the use of NVMe over Fabrics for front end client IO, both NVMe/FC and NVMe/TCP.
- **FlexClone** – Enables rapid creation of space efficient cloning of data based on snapshots.
- **S3** – Enables the S3 protocol for front end client IO.
- **SnapRestore** – Enables rapid recovery of data from snapshots.
- **Autonomous Ransomware Protection** - Enables the automatic protection of NAS file shares when abnormal filesystem activity is detected.
- **Multi Tenant Key Manager** - Enables the ability to have multiple key managers for different tenants on the system.
- **SnapLock** – Enables the protection of data from modification, deletion or corruption on the system.
- **SnapMirror Cloud** – Enables the replication of system volumes to object targets.
- **S3 SnapMirror** – Enables the replication of ONTAP S3 objects to alternate S3 compatible targets.

NetApp All-Flash SAN Array

The NetApp All-Flash SAN Array (ASA) is a high-performance storage solution designed to meet the demanding requirements of modern data centers. It combines the speed and reliability of flash storage with NetApp’s advanced data management features to deliver exceptional performance, scalability, and data protection.

The ASA lineup is comprised of both A-Series and C-Series models.

The NetApp A-Series all-NVMe flash arrays are designed for high-performance workloads, offering ultra-low latency and high resiliency, making them suitable for mission-critical applications.
C-Series QLC flash arrays are aimed at higher-capacity use cases, delivering the speed of flash with the economy of hybrid flash.

For detailed information see the NetApp ASA landing page.

NetApp ASA features

The NetApp All-Flash SAN Array includes the following features:

**Performance** - The All-Flash SAN Array leverages solid-state drives (SSDs), with an end-to-end NVMe architecture, to provide lightning-fast performance, significantly reducing latency and improving application response times. It delivers consistent high IOPS and low latency, making it suitable for latency-sensitive workloads such as databases, virtualization, and analytics.

**Scalability** - NetApp All-Flash SAN Arrays are built with a scale-out architecture, allowing organizations to seamlessly scale their storage infrastructure as their needs grow. With the ability to add additional storage nodes, organizations can expand capacity and performance without disruption, ensuring that their storage can keep up with increasing data demands.

**Data Management** - NetApp’s Data ONTAP operating system powers the All-Flash SAN Array, providing a comprehensive suite of data management features. These include thin provisioning, deduplication, compression, and data compaction, which optimize storage utilization and reduce costs. Advanced data protection features like snapshots, replication, and encryption ensure the integrity and security of stored data.

**Integration and Flexibility** - The All-Flash SAN Array integrates with NetApp’s broader ecosystem, enabling seamless integration with other NetApp storage solutions, such as hybrid cloud deployments with NetApp Cloud Volumes ONTAP. It also supports industry-standard protocols like Fibre Channel (FC) and iSCSI, enabling easy integration into existing SAN infrastructures.

**Analytics and Automation** - NetApp’s management software, including NetApp Cloud Insights, provides comprehensive monitoring, analytics, and automation capabilities. These tools enable administrators to gain
insights into their storage environment, optimize performance, and automate routine tasks, simplifying storage management and improving operational efficiency.

**Data Protection and Business Continuity** - The All-Flash SAN Array offers built-in data protection features such as point-in-time snapshots, replication, and disaster recovery capabilities. These features ensure data availability and facilitate rapid recovery in the event of data loss or system failures.

**Protocol Support**

The ASA supports all standard SAN protocols including, iSCSI, Fibre Channel (FC), Fibre Channel over Ethernet (FCoE), and NVME over fabrics.

**iSCSI** - NetApp ASA provides robust support for iSCSI, allowing block-level access to storage devices over IP networks. It offers seamless integration with iSCSI initiators, enabling efficient provisioning and management of iSCSI LUNs. ONTAP’s advanced features, such as multi-pathing, CHAP authentication, and ALUA support.

For design guidance on iSCSI configurations refer to .

**Fibre Channel** - NetApp ASA offers comprehensive support for Fibre Channel (FC), a high-speed network technology commonly used in storage area networks (SANs). ONTAP seamlessly integrates with FC infrastructure, providing reliable and efficient block-level access to storage devices. It offers features like zoning, multi-pathing, and fabric login (FLOGI) to optimize performance, enhance security, and ensure seamless connectivity in FC environments.

For design guidance on Fibre Channel configurations refer to the SAN Configuration reference documentation.

**NVMe over Fabrics** - NetApp ONTAP and ASA support NVMe over fabrics. NVMe/FC enables the use of NVMe storage devices over Fibre Channel infrastructure, and NVMe/TCP over storage IP networks.

For design guidance on NVMe refer to NVMe configuration, support and limitations.

**Active-active technology**

NetApp All-Flash SAN Arrays allows for active-active paths through both controllers, eliminating the need for the host operating system to wait for an active path to fail before activating the alternative path. This means that the host can utilize all available paths on all controllers, ensuring active paths are always present regardless of whether the system is in a steady state or undergoing a controller failover operation.

Furthermore, the NetApp ASA offers a distinctive feature that greatly enhances the speed of SAN failover. Each controller continuously replicates essential LUN metadata to its partner. As a result, each controller is prepared to take over data serving responsibilities in the event of a sudden failure of its partner. This readiness is possible because the controller already possesses the necessary information to start utilizing the drives that were previously managed by the failed controller.

With active-active pathing, both planned and unplanned takeovers have IO resumption times of 2-3 seconds.

For more information see TR-4968, NetApp All-SAS Array – Data Availability and Integrity with the NetApp ASA.

**Storage guarantees**

NetApp offers a unique set of storage guarantees with NetApp All-flash SAN Arrays. The unique benefits include:

**Storage efficiency guarantee:** Achieve high performance while minimizing storage cost with the Storage Efficiency Guarantee. 4:1 for SAN workloads.
6 Nines (99.9999%) data availability guarantee: Guarantees remediation for unplanned downtime in excess of 31.56 seconds per year.

Ransomware recovery guarantee: Guaranteed data recovery in the event of a ransomware attack.

See the NetApp ASA product portal for more information.

---

### NetApp Plug-ins for VMware vSphere

NetApp storage services are tightly integrated with VMware vSphere through the use of the following plug-ins:

#### ONTAP Tools for VMware vSphere

The ONTAP Tools for VMware allows administrators to manage NetApp storage directly from within the vSphere Client. ONTAP Tools allows you to deploy and manage datastores, as well as provision vVol datastores. ONTAP Tools allows mapping of datastores to storage capability profiles which determine a set of storage system attributes. This allows the creation of datastores with specific attributes such as storage performance and QoS.

ONTAP Tools includes the following components:

- **Virtual Storage Console (VSC):** The VSC includes the interface integrated with the vSphere client where you can add storage controllers, provision datastores, monitor performance of datastores, and view and update ESXi host settings.

- **VASA Provider:** The VMware vSphere APIs for Storage Awareness (VASA) Provider for ONTAP send information about storage used by VMware vSphere to the vCenter Server, enabling provisioning of VMware Virtual Volumes (vVols) datastores, creation and use of storage capability profiles, compliance verification, and performance monitoring.

- **Storage Replication Adapter (SRA):** When enabled and used with VMware Site Recovery Manager (SRM), SRA facilitates the recovery of vCenter Server datastores and virtual machines in the event of a failure, allowing configuration of protected sites and recovery sites for disaster recovery.

For more information on NetApp ONTAP tools for VMware see [ONTAP tools for VMware vSphere Documentation](#).

#### SnapCenter Plug-in for VMware vSphere

The SnapCenter Plug-in for VMware vSphere (SCV) is a software solution from NetApp that offers comprehensive data protection for VMware vSphere environments. It is designed to simplify and streamline the process of protecting and managing virtual machines (VMs) and datastores.

The SnapCenter Plug-in for VMware vSphere provides the following capabilities in a unified interface, integrated with the vSphere client:

- **Policy-Based Snapshots** - SnapCenter allows you to define policies for creating and managing application-consistent snapshots of virtual machines (VMs) in VMware vSphere.

- **Automation** - Automated snapshot creation and management based on defined policies help ensure consistent and efficient data protection.

---
**VM-Level Protection** - Granular protection at the VM level allows for efficient management and recovery of individual virtual machines.

**Storage Efficiency Features** - Integration with NetApp storage technologies provides storage efficiency features like deduplication and compression for snapshots, minimizing storage requirements.

The SnapCenter Plug-in orchestrates the quiescing of virtual machines in conjunction with hardware-based snapshots on NetApp storage arrays. SnapMirror technology is utilized to replicate copies of backups to secondary storage systems including in the cloud.

For more information refer to the [SnapCenter Plug-in for VMware vSphere documentation](#).

BlueXP integration enables 3-2-1 backup strategies that extend copies of data to object storage in the cloud.

For more information on 3-2-1 backup strategies with BlueXP visit [3-2-1 Data Protection for VMware with SnapCenter Plug-in and BlueXP backup and recovery for VMs](#).

**NetApp Cloud Insights**

NetApp Cloud Insights simplifies observation of on-prem and cloud infrastructure and provides analytics and troubleshooting capabilities to help solve complex problems. Cloud Insights works by collecting data from a data center environment and sending that data to the cloud. This is done with locally installed software called an Acquisition Unit and with specific collectors enabled for the assets in the data center.

The assets in Cloud Insights can be tagged with annotations that provide a method of organizing and classifying data. Dashboard can be created using a wide variety of widgets for displaying the data and Metric Queries can be created for detailed tabular views of data.

Cloud Insights comes with a large number of ready-made dashboards that help to zero in on specific types of problem areas and categories of data.

Cloud Insights is a heterogeneous tool designed to collect data from a wide range of devices. However, there is a library of templates, called ONTAP Essentials, that makes it easy for NetApp customers to get started quickly.

For detailed information on how to get started with Cloud Insights refer to the [NetApp BlueXP and Cloud Insights landing page](#).

**NetApp All-Flash SAN Array with VMware vSphere 8**

Author: Josh Powell - NetApp Solutions Engineering

**Managing Block Storage with ONTAP Tools for VMware vSphere**

The ONTAP Tools for VMware allows administrators to manage NetApp storage directly from within the vSphere Client. ONTAP Tools allows you to deploy and manage datastores, as well as provision vVol datastores.

ONTAP Tools allows mapping of datastores to storage capability profiles which determine a set of storage system attributes. This allows the creation of datastores with specific attributes such as storage performance and QoS.

ONTAP Tools includes the following components:
**Virtual Storage Console (VSC):** The VSC includes the interface integrated with the vSphere client where you can add storage controllers, provision datastores, monitor performance of datastores, and view and update ESXi host settings.

**VASA Provider:** The VMware vSphere APIs for Storage Awareness (VASA) Provider for ONTAP send information about storage used by VMware vSphere to the vCenter Server, enabling provisioning of VMware Virtual Volumes (vVols) datastores, creation and use of storage capability profiles, compliance verification, and performance monitoring.

**Storage Replication Adapter (SRA):** When enabled and used with VMware Site Recovery Manager (SRM), SRA facilitates the recovery of vCenter Server datastores and virtual machines in the event of a failure, allowing configuration of protected sites and recovery sites for disaster recovery.

For more information on NetApp ONTAP tools for VMware see [ONTAP tools for VMware vSphere Documentation](#).

**Solution Deployment Overview**

In this solution we will demonstrate the use of the ONTAP Tools for VMware vSphere to provision a VMware Virtual Volumes (vVol) datastores and create a virtual machine on a vVol datastore.

In a vVols datastore each virtual disk is a vVol and becomes a native LUN object on the storage system. The integration of the storage system and vSphere takes place through the VMware API’s for Storage Awareness (VASA) provider (installed with ONTAP Tools) and allows the storage system to be aware of the VM data and manage it accordingly. Storage policies, defined in the vCenter Client are used to allocate and manage storage resources.

For detailed information on vVols with ONTAP refer to [Virtual Volumes vVols) with ONTAP](#).

This solution covers the following high level steps:

1. Add a storage system in ONTAP Tools.
2. Create a storage capability profile in ONTAP Tools.
3. Create a vVols datastore in ONTAP Tools.
4. Create a VM storage policy in the vSphere client.
5. Create a new virtual machine on the vVol datastore.

**Prerequisites**

The following components were used in this solution:

2. iSCSI SVM created on the ASA with network connectivity to the ESXi hosts.
3. ONTAP Tools for VMware vSphere 9.13 (VASA provider enabled by default).
4. vSphere 8.0 cluster (vCenter appliance, and ESXi hosts).

**Solution Deployment**

**Create a vVols datastore in ONTAP Tools**

To create a vVols datastore in ONTAP Tools complete the following steps:
Add a storage system to ONTAP Tools.

1. Access NetApp ONTAP Tools by selecting it from the main menu in the vSphere client.

2. In ONTAP Tools select **Storage Systems** from the left hand menu and then press **Add**.
3. Fill out the IP Address, credentials of the storage system and the port number. Click on Add to start the discovery process.

Add Storage System

Any communication between ONTAP tools plug-in and the storage system should be mutually authenticated.

- **vCenter server**: 10.61.181.205
- **Name or IP address**: 10.192.102.103
- **Username**: admin
- **Password**: ********
- **Port**: 443
- **ONTAP Cluster Certificate**: Automatically fetch

**Add**
Create a storage capability profile in ONTAP Tools

Storage capability profiles describe the features provided by a storage array or storage system. They include quality of service definitions and are used to select storage systems that meet the parameters defined in the profile.

To create a storage capability profile in ONTAP Tools complete the following steps:

1. In ONTAP Tools select **Storage capability profile** from the left hand menu and then press **Create**.

2. In the **Create Storage Capability profile** wizard provide a name and description of the profile and click on **Next**.

3. Select the platform type and to specify the storage system is to be an All-Flash SAN Array set **Asymmetric** to false.
4. Next, select choice of protocol or Any to allow all possible protocols. Click Next to continue.

5. The performance page allows setting of quality of service in form of minimum and maximum IOPs allowed.
6. Complete the **storage attributes** page selecting storage efficiency, space reservation, encryption and any tiering policy as needed.

7. Finally, review the summary and click on Finish to create the profile.
<table>
<thead>
<tr>
<th>Summary</th>
<th>ASA_Gold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>N/A</td>
</tr>
<tr>
<td>Description</td>
<td>Performance</td>
</tr>
<tr>
<td>Platform</td>
<td>No</td>
</tr>
<tr>
<td>Asymmetric</td>
<td>Any</td>
</tr>
<tr>
<td>Protocol</td>
<td>6000 IOPS</td>
</tr>
<tr>
<td>Max IOPS</td>
<td>Thin</td>
</tr>
<tr>
<td>Space reserve</td>
<td>Yes</td>
</tr>
<tr>
<td>Deduplication</td>
<td>Yes</td>
</tr>
<tr>
<td>Compression</td>
<td>No</td>
</tr>
<tr>
<td>Encryption</td>
<td>None</td>
</tr>
<tr>
<td>Tiering policy (FabricPool)</td>
<td>None</td>
</tr>
</tbody>
</table>
To create a vVols datastore in ONTAP Tools complete the following steps:

1. In ONTAP Tools select **Overview** and from the **Getting Started** tab click on **Provision** to start the wizard.

2. On the **General** page of the New Datastore wizard select the vSphere datacenter or cluster destination. Select **vVols** as the datastore type, fill out a name for the datastore, and select the protocol.

3. On the **Storage system** page select the select a storage capability profile, the storage system and SVM. Click on **Next** to continue.
4. On the Storage attributes page select to create a new volume for the datastore and fill out the storage attributes of the volume to be created. Click on Add to create the volume and then Next to continue.

5. Finally, review the summary and click on Finish to start the vVol datastore creation process.
Create a VM storage policy in the vSphere client

A VM storage policy is a set of rules and requirements that define how virtual machine (VM) data should be stored and managed. It specifies the desired storage characteristics, such as performance, availability, and data services, for a particular VM.

In this case, the task involves creating a VM storage policy to specify that a virtual machine will be generated on vVol datastores and to establish a one-to-one mapping with the previously generated storage capability profile.
Create a VM storage policy

To create a VM storage policy complete the following steps:

1. From the vSphere clients main menu select Policies and Profiles.

2. In the Create VM Storage Policy wizard, first fill out a name and description for the policy and click on Next to continue.

4. On the next page specific to the policy structure chosen, select the storage capability profile that describes the storage system(s) to be used in the VM storage policy. Click on Next to continue.

5. On the Storage compatibility page, review the list of vSAN datastores that match this policy and click Next.

6. Finally, review the policy to be implemented and click on Finish to create the policy.

Create a VM storage policy in the vSphere client

A VM storage policy is a set of rules and requirements that define how virtual machine (VM) data should be stored and managed. It specifies the desired storage characteristics, such as performance, availability, and data services, for a particular VM.

In this case, the task involves creating a VM storage policy to specify that a virtual machine will be generated
on vVol datastores and to establish a one-to-one mapping with the previously generated storage capability profile.
Create a virtual machine on a vVol datastore

The final step is to create a virtual machine using the VM storage policies previously created:

1. From the **New Virtual Machine** wizard select **Create a new virtual machine** and select **Next** to continue.

2. Fill in a name and select a location for the virtual machine and click on **Next**.

3. On the **Select a compute resource** page select a destination and click on **Next**.

4. On the **Select storage** page select a VM Storage Policy and the vVols datastore that will be the destination for the VM. Click on **Next**.
5. On the **Select compatibility** page choose the vSphere version(s) that the VM will be compatible with.

6. Select the guest OS family and version for the new VM and click on **Next**.

7. Fill out the **Customize hardware** page. Note that a separate VM storage policy can be selected for each hard disk (VMDK file).
8. Finally, review the summary page and click on Finish to create the VM.

In summary, NetApp ONTAP Tools automates the process of creating vVol datastores on ONTAP storage systems. Storage capability profiles define not only the storage systems to be used for datastore creation but also dictate QoS policies that can be implemented on an individual VMDK basis. vVols provide a simplified storage management paradigm and tight integration between NetApp and VMware make this a practical solution for streamlined, efficient, and granular control over virtualized environments.

**NetApp All-Flash SAN Array with VMware vSphere 8**

Author: Josh Powell - NetApp Solutions Engineering

**Monitoring On-Premises Storage with NetApp Cloud Insights**

NetApp Cloud Insights is a cloud-based infrastructure monitoring and analytics platform designed to provide comprehensive visibility and insights into the performance, health, and costs of IT infrastructures, both on-premises and in the cloud. Key features of NetApp Cloud Insights include real-time monitoring, customizable dashboards, predictive analytics, and cost optimization tools, allowing organizations to effectively manage and optimize their on-premises and cloud environments.

NetApp Cloud Insights operates through Acquisition Unit software, which is set up with data collectors for assets such as VMware vSphere and NetApp ONTAP storage systems. These collectors gather data and transmit it to Cloud Insights. The platform then utilizes a variety of dashboards, widgets, and metric queries to organize the data into insightful analyses for users to interpret.

Cloud Insights architecture diagram:
**Solution Deployment Overview**

This solution provides an introduction to monitoring on-premises VMware vSphere and ONTAP storage systems using NetApp Cloud Insights.

This list provides the high level steps covered in this solution:

1. Configure Data Collector for a vSphere cluster.
2. Configure Data Collector for an ONTAP storage system.
3. Use Annotation Rules to tag assets.
4. Explore and correlate assets.
5. Use a Top VM Latency dashboard to isolate noisy neighbors.
6. Identify opportunities to rightsize VMs.
7. Use queries to isolate and sort metrics.

**Prerequisites**

This solution uses the following components:

2. VMware vSphere 8.0 cluster.
4. NetApp Cloud Insights Acquisition Unit software installed on a local VM with network connectivity to assets for data collection.
Solution Deployment

Configure Data Collectors

To configure Data Collectors for VMware vSphere and ONTAP storage systems complete the following steps:
Add a Data Collector for an ONTAP storage systems

1. Once logged into Cloud Insights, navigate to Observability > Collectors > Data Collectors and press the button to install a new Data Collector.

2. From here search for ONTAP and click on ONTAP Data Management Software.

3. On the Configure Collector page fill out a name for the collector, specify the correct Acquisition Unit and provide the credentials for the ONTAP storage system. Click on Save and Continue and then Complete Setup at the bottom of the page to complete the configuration.
Add a Data Collector for a VMware vSphere cluster

1. Once again, navigate to **Observability > Collectors > Data Collectors** and press the button to install a new Data Collector.

2. From here search for **vSphere** and click on **VMware vSphere**.

3. On the **Configure Collector** page fill out a name for the collector, specify the correct **Acquisition Unit** and provide the credentials for the vCenter server. Click on **Save and Continue** and then **Complete Setup** at the bottom of the page to complete the configuration.
Add Annotations to assets

Annotations are a useful method of tagging assets so that they can be filtered and otherwise identified in the various views and metric queries available in Cloud Insights.

In this section, annotations will be added to virtual machine assets for filtering by Data Center.
Use Annotation Rules to tag assets

1. In the left-hand menu, navigate to **Observability > Enrich > Annotation Rules** and click on the **+ Rule** button in the upper right to add a new rule.

2. In the **Add Rule** dialog box fill in a name for the rule, locate a query to which the rule will be applied, the annotation field affected, and the value to be populated.
3. Finally, in the upper right hand corner of the **Annotation Rules** page click on **Run All Rules** to run the rule and apply the annotation to the assets.

**Explore and correlate assets**

Cloud Insights draws logical conclusions about the assets that are running together on your storage systems and vsphere clusters.

This sections illustrates how to use dashboards to correlate assets.
Correlating assets from a storage performance Dashboard

1. In the left-hand menu, navigate to **Observability > Explore > All Dashboards**.

2. Click on the **+ From Gallery** button to view a list of ready-made dashboards that can be imported.

3. Choose a dashboard for FlexVol performance from the list and click on the **Add Dashboards** button at the bottom of the page.
4. Once imported, open the dashboard. From here you can see various widgets with detailed performance data. Add a filter to view a single storage system and select a storage volume to drill into its details.

5. From this view you can see various metrics related to this storage volume and the top utilized and correlated virtual machines running on the volume.
6. Clicking on the VM with the highest utilization drills into the metrics for that VM to view any potential issues.

Use Cloud Insights to identify noisy neighbors

Cloud Insights features dashboards that can easily isolate peer VMs that are negatively impacting other VMs running on the same storage volume.
Use a Top VM Latency dashboard to isolate noisy neighbors

1. In this example access a dashboard available in the **Gallery** called **VMware Admin - Where do I have VM Latency?**

   ![Dashboard Groups (108) My Dashboards (6)](image)

   ![VMware Admin - Where do I have VM Latency?](image)

2. Next, filter by the **Data Center** annotation created in a previous step to view a subset of assets.

3. This dashboard shows a list of the top 10 VMs by average latency. From here click on the VM of concern to drill into its details.
4. The VMs potentially causing workload contention are listed and available. Drill into these VMs performance metrics to investigate any potential issues.
View over and under utilized resources in Cloud Insights

By matching VM resources to actual workload requirements, resource utilization can be optimized, leading to cost savings on infrastructure and cloud services. Data in Cloud Insights can be customized to easily display over or under utilized VMs.
Identify opportunities to right size VMs

1. In this example access a dashboard available in the Gallery called **VMware Admin - Where are opportunities to right size?**


<table>
<thead>
<tr>
<th>My Dashboards (6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name ↑</td>
</tr>
<tr>
<td>All SAN Array Status (2)</td>
</tr>
<tr>
<td>Cloud Volumes ONTAP - FlexVol Performance (6)</td>
</tr>
<tr>
<td>ONTAP - Volume Workload Performance (Frontend) (7)</td>
</tr>
<tr>
<td><strong>VMware Admin - Where are opportunities to right size?</strong> (37)</td>
</tr>
<tr>
<td>VMware Admin - Where do I potentially reclaim waste? (11)</td>
</tr>
<tr>
<td>VMware Admin - Where do I have VM Latency? (9)</td>
</tr>
</tbody>
</table>

2. First filter by all of the ESXi hosts in the cluster. You can then see ranking of the top and bottom VMs by memory and CPU utilization.
3. Tables allow sorting and provide more detail based on the columns of data chosen.
4. Another dashboard called **VMware Admin - Where can I potentially reclaim waste?** shows powered off VM's sorted by their capacity use.
Use queries to isolate and sort metrics

The amount of data captured by Cloud Insights is quite comprehensive. Metric queries provide a powerful way to sort and organize large amounts of data in useful ways.
View a detailed VMware query under ONTAP Essentials

1. Navigate to **ONTAP Essentials > VMware** to access a comprehensive VMware metric query.

2. In this view you are presented with multiple options for filtering and grouping the data at the top. All columns of data are customizable and additional columns can be easily added.
Conclusion

This solution was designed as a primer to learn how to get started with NetApp Cloud Insights and show some of the powerful capabilities that this observability solution can provide. There are hundreds of dashboards and metric queries built into the product which makes it easy to get going immediately. The full version of Cloud Insights is available as a 30-day trial and the basic version is available free to NetApp customers.

Additional Information

To learn more about the technologies presented in this solution refer to the following additional information.

- NetApp BlueXP and Cloud Insights landing page
- NetApp Cloud Insights documentation

Demos and Tutorials

Virtualization Videos and Demos

See the following videos and demos highlighting specific features of the hybrid cloud, virtualization, and container solutions.

NetApp ONTAP Tools for VMware vSphere

- ONTAP Tools for VMware - Overview
- VMware iSCSI Datastore Provisioning with ONTAP
- VMware NFS Datastore Provisioning with ONTAP
VMware Cloud on AWS with AWS FSx for NetApp ONTAP

- Windows Guest Connected Storage with FSx ONTAP using iSCSI
- Linux Guest Connected Storage with FSx ONTAP using NFS
- VMware Cloud on AWS TCO savings with Amazon FSx for NetApp ONTAP
- VMware Cloud on AWS supplemental datastore w/ Amazon FSx for NetApp ONTAP
- VMware HCX Deployment and Configuration Setup for VMC
- vMotion Migration Demonstration with VMware HCX for VMC and FSxN
- Cold Migration Demonstration with VMware HCX for VMC and FSxN

Azure VMware Services on Azure with Azure NetApp Files (ANF)

- Azure VMware Solution supplemental datastore overview with Azure NetApp Files
- Azure VMware Solution DR with Cloud Volumes ONTAP, SnapCenter and JetStream
- Cold Migration Demonstration with VMware HCX for AVS and ANF
- vMotion Demonstration with VMware HCX for AVS and ANF
- Bulk Migration Demonstration with VMware HCX for AVS and ANF

SnapCenter Plug-in for VMware vSphere

NetApp SnapCenter software is an easy-to-use enterprise platform to securely coordinate and manage data protection across applications, databases, and file systems.

The SnapCenter Plug-in for VMware vSphere allows you to perform backup, restore, and attach operations for VMs and backup and mount operations for datastores that are registered with SnapCenter directly within VMware vCenter.

For more information about NetApp SnapCenter Plug-in for VMware vSphere, see the NetApp SnapCenter Plug-in for VMware vSphere Overview.

- SnapCenter Plug-in for VMware vSphere - Solution Pre-Requisites
- SnapCenter Plug-in for VMware vSphere - Deployment
- SnapCenter Plug-in for VMware vSphere - Backup Workflow
- SnapCenter Plug-in for VMware vSphere - Restore Workflow
- SnapCenter - SQL Restore Workflow
3-2-1 Data Protection Solutions

3-2-1 data protection solutions combine on-premises primary and secondary backups, using SnapMirror technology, with replicated copies to object storage using BlueXP backup and recovery.

3-2-1 Data Protection for VMFS Datastores with SnapCenter Plug-in for VMware vSphere and BlueXP Backup and Recovery for Virtual Machines
VMware Tanzu enables customers to deploy, administer, and manage their Kubernetes environment through vSphere or the VMware Cloud Foundation. This portfolio of products from VMware allows customer to manage all their relevant Kubernetes clusters from a single control plane by choosing the VMware Tanzu edition that best suits their needs.

For more information about VMware Tanzu, see the VMware Tanzu Overview. This review covers use cases, available additions, and more about VMware Tanzu.
NetApp Cloud Insights

NetApp Cloud Insights is comprehensive monitoring and analytics platform designed to provide visibility and control over your on-premises and cloud infrastructure.

NetApp Cloud Insights - Observability for the Modern Datacenter
Solution Automation

NetApp Solution Automation

Introduction

In providing solutions to meet today’s business challenges, NetApp delivers solutions with the following goals:

• Providing validated deployment and configuration steps,
• Providing solutions that are easily consumable,
• Providing solution deployment that has a predictable outcome, is easily repeated, and scalable across a customer’s enterprise.

In order to achieve these goals, it is paramount that the deployment and configuration of infrastructure and/or applications delivered through our solutions is simplified through automation. NetApp is committed to simplifying solution consumption through automation.

Utilizing open-source automation tools such as Red Hat Ansible, HashiCorp Terraform, or Microsoft Powershell, NetApp solutions have the ability to automate application deployment, cloud provisioning, configuration management, and many other common IT tasks. NetApp’s solutions take advantage of publicly available automation artifacts - as well as providing NetApp authored automation - to simplify the overall deployment of a solution.

Where automation capabilities are available, the solution collateral will guide the user through the process for automating the solution or solution steps via the specific automation tool(s).

Getting Started with NetApp solution automation

NetApp solution automation provides simplicity and repeatability for many of the common tasks utilized by the NetApp Solutions.

Prior to running any solution automation, the environment must be configured for how the automation will be executed. There are options to run the automation from the command line or through a tool such as AWX or tower.

The following sections will outline the steps required to configure the environment for each of the specified environments.
Setup the Ansible Control Node for CLI deployments on RHEL / CentOS

1. Requirements for the Ansible control node:
   a. A RHEL/CentOS machine with the following packages installed:
      i. Python3
      ii. Pip3
      iii. Ansible (version greater than 2.10.0)
      iv. Git

If you have a fresh RHEL/CentOS machine without the above requirements installed, follow the below steps to setup that machine as the Ansible control node:

1. Enable the Ansible repository for RHEL-8/RHEL-7
   a. For RHEL-8 (run the below command as root)

```
subscription-manager repos --enable ansible-2.9-for-rhel-8-x86_64-rpms
```

   b. For RHEL-7 (run the below command as root)

```
subscription-manager repos --enable rhel-7-server-ansible-2.9-rpms
```

2. Paste the below content in the Terminal

```
sudo yum -y install python3 >> install.log
sudo yum -y install python3-pip >> install.log
python3 -W ignore -m pip --disable-pip-version-check install ansible >> install.log
sudo yum -y install git >> install.log
```
Setup the Ansible Control Node for CLI deployments on Ubuntu / Debian

1. Requirements for the Ansible control node,:  
   a. A Ubuntu/Debian machine with the following packages installed:  
      i. Python3  
      ii. Pip3  
      iii. Ansible (version greater than 2.10.0)  
      iv. Git

If you have a fresh Ubuntu/Debian machine without the above requirements installed, follow the below steps to setup that machine as the Ansible control node:

1. Paste the below content in the terminal

```
sudo apt-get -y install python3 >> outputlog.txt
sudo apt-get -y install python3-pip >> outputlog.txt
python3 -W ignore --disable-pip-version-check install ansible
>> outputlog.txt
sudo apt-get -y install git >> outputlog.txt
```
Setup Ansible Tower or AWX for Tower / AWX deployments

This section describes the steps required to configure the parameters in AWX/Ansible Tower that prepare the environment for consuming NetApp automated solutions.

1. Configure the inventory.
   a. Navigate to Resources → Inventories → Add and click Add Inventory.
   b. Provide name and organization details and click Save.
   c. In the Inventories page, click the inventory resources you just created.
   d. If there are any inventory variables, paste them into the variables field.
   e. Go to the Groups sub-menu and click Add.
   f. Provide the name of the group, copy in the group variables (if necessary), and click Save.
   g. Click the group created, go to the Hosts sub-menu and click Add New Host.
   h. Provide the hostname and IP address of the host, paste in the host variables (if necessary), and click Save.

2. Create credential types. For solutions involving ONTAP, Element, VMware, or any other HTTPS-based transport connection, you must configure the credential type to match the username and password entries.
   a. Navigate to Administration → Credential Types and click Add.
   b. Provide the name and description.
   c. Paste the following content into the Input Configuration:

```
fields:
- id: username
  type: string
  label: Username
- id: password
  type: string
  label: Password
  secret: true
- id: vsadmin_password
  type: string
  label: vsadmin_password
  secret: true
```

   a. Paste the following content into the Injector Configuration:

```
extra_vars:
  password: '{{ password }}'
  username: '{{ username }}'
  vsadmin_password: '{{ vsadmin_password }}'
```
1. Configure credentials.
   a. Navigate to Resources → Credentials and click Add.
   b. Enter the name and organization details.
   c. Select the correct credential type; if you intend to use the standard SSH login, select the type Machine or alternatively select the custom credential type that you created.
   d. Enter the other corresponding details and click Save.

2. Configure the project.
   a. Navigate to Resources → Projects and click Add.
   b. Enter the name and organization details.
   c. Select Git for the Source Control Credential Type.
   d. Paste the source control URL (or git clone URL) corresponding to the specific solution.
   e. Optionally, if the Git URL is access controlled, create and attach the corresponding credential in Source Control Credential.
   f. Click Save.

3. Configure the job template.
   a. Navigate to Resources → Templates → Add and click Add Job Template.
   b. Enter the name and description.
   c. Select the Job type; Run configures the system based on a playbook and Check performs a dry run of the playbook without actually configuring the system.
   d. Select the corresponding inventory, project, and credentials for the playbook.
   e. Select the playbook that you would like to run as a part of the job template.
   f. Usually the variables are pasted during runtime. Therefore, to get the prompt to populate the variables during runtime, make sure to tick the checkbox Prompt on Launch corresponding to the Variable field.
   g. Provide any other details as required and click Save.

4. Launch the job template.
   a. Navigate to Resources → Templates.
   b. Click the desired template and then click Launch.
   c. Fill in any variables if prompted on launch and then click Launch again.

---

NetApp Solution Automation

AWS Authentication Requirements for CVO and Connector Using NetApp Cloud Manager

To configure automated Deployments of CVO and Connectors using Ansible playbooks via AWX/Ansible Tower, the following information is needed:

**Acquiring Access/Secret Keys from AWS**

1. To deploy CVO and Connector in Cloud Manager, we need AWS Access/Secret Key. Acquire the keys in AWS console by launching IAM→Users→your username→security credentials→Create Access key.
2. Copy access keys and keep them secured to use in Connector and CVO deployment.

- If you lose your key, you can create another access key and delete the one you lost

**Acquiring Refresh Token from NetApp Cloud Central**

1. Login into your cloud central account using your account credentials at [https://services.cloud.netapp.com/refresh-token](https://services.cloud.netapp.com/refresh-token)

2. Generate a refresh Token and save it for deployments.

**Refresh Token Generator**

You can use this refresh token to obtain an access tokens for users. Store this refresh token securely. If necessary, you can revoke the token at a later time by navigating to the Refresh Token Generator.

Note that this token is displayed on this page only—it is not stored on our servers. The token will no longer be displayed if you refresh or leave this page.

REFRESH TOKEN: EAAfPTMCuu4Ql9hR2PTRT75Lswr0fH4BheJEiT2XFsHt

**Acquiring Client ID**

1. Access the API page to copy Client ID at [https://services.cloud.netapp.com/developer-hub](https://services.cloud.netapp.com/developer-hub).

2. Click on "learn How to Authenticate", in the top right corner.

3. From the Authentication window that pops up, copy the Client ID from Regular Access if you require a username/password to login. Federated users with SSO should copy the client ID from the "Refresh Token Tab".
Acquiring Key Pair from AWS

1. In AWS console, search for “Key Pair” and create a key pair with “pem”. Remember the name of your key_pair, we will use it to deploy the connector.

Authentication Information

NetApp Cloud Central Services use OAuth 2.0, an industry-standard protocol, for authorization.

Communicating with an authenticated endpoint is a two-step process:
1. Acquire a JWT access token from the OAuth token endpoint.
2. Call an API endpoint with the JWT access token.

Non-federated users can use regular access or refresh token access, federated users must use refresh token access.

| Regular Access | Refresh Token Access (Required for federated users) |

How to Acquire a JWT Access Token via regular token access

1. Make an HTTP POST request to the endpoint
   https://netapp-cloud-account.acthe.com/oauth/token

   Include the header: Content-Type: application/json

   Include the body:

   ```json
   {
     "grant_type": "password",
     "username": "YOUR_EMAIL_ADDRESS",
     "password": "YOUR_PASSWORD",
     "audience": "https://api.cloud.netapp.com",
     "client_id": "",
   }
   ```

Acquiring Key Pair from AWS

1. In AWS console, search for “Key Pair” and create a key pair with “pem”. Remember the name of your key_pair, we will use it to deploy the connector.
Acquiring Account ID

1. In Cloud Manager, click on Account -> Manage Accounts and then copy the account id for use in variables for AWX.

```
account-lwO1qR76
```

Account ID

---

Cloud Volumes Automation via Terraform

This solution documents the automated deployments of Cloud Volumes on AWS (CVO Single Node, CVO HA and FSX ONTAP) and Azure (CVO Single Node, CVO HA and ANF) using Terraform modules. The code can be found at https://github.com/NetApp/na_cloud_volumes_automation

Pre-requisites

1. Terraform >= 0.13
2. Cloud Manager Account
3. Cloud Provider Account – AWS, Azure
4. Host machine (any OS supported by Terraform)

Provider documentation

The documentation of Terraform provider for Cloud Manager is available at: https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs

Controlling the provider version

Note that you can also control the provider version. This is controlled by a required_providers block in your Terraform configuration.

The syntax is as follows:

```terraform
terraform {
  required_providers {
    netapp-cloudmanager = {
      source = "NetApp/netapp-cloudmanager"
      version = ">= 23.0.0"
    }
  }
}
```
Read more on provider version control.

Running Specific Modules
CVO Single Node Deployment

Terraform configuration files for deployment of NetApp CVO (Single Node Instance) on AWS

This section contains various Terraform configuration files to deploy/configure single node NetApp CVO (Cloud Volumes ONTAP) on AWS (Amazon Web Services).

Terraform Documentation: https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs

Procedure

In order to run the template:

a. Clone the repository.
   
   ```
   git clone https://github.com/NetApp/na_cloud_volumes_automation.git
   ```

b. Navigate to the desired folder
   
   ```
   cd na_cloud_volumes_automation/
   ```

c. Configure AWS credentials from the CLI.
   
   ```
   aws configure
   ```
   - AWS Access Key ID [None]: accesskey
   - AWS Secret Access Key [None]: secretkey
   - Default region name [None]: us-west-2
   - Default output format [None]: json

d. Update the variable values in `vars/aws_cvo_single_node_deployment.tfvar`
   
   You can choose to deploy the connector by setting the variable "aws_connector_deploy_bool" value to true/false.

e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.
   
   ```
   terraform init
   ```

f. Verify the terraform files using terraform validate command.
   
   ```
   terraform validate
   ```
g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

```terraform
terraform plan -target="module.aws_sn" -var -file="vars/aws_cvo_single_node_deployment.tfvars"
```

h. Run the deployment

```terraform
terraform apply -target="module.aws_sn" -var -file="vars/aws_cvo_single_node_deployment.tfvars"
```

To delete the deployment

```terraform
terraform destroy
```

Recipies:
Connector

Terraform variables for NetApp AWS connector instance for CVO deployment.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aws_connector_deploy_bool</td>
<td>Bool</td>
<td>(Required) Check for Connector deployment.</td>
</tr>
<tr>
<td>aws_connector_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Manager Connector.</td>
</tr>
<tr>
<td>aws_connector_region</td>
<td>String</td>
<td>(Required) The region where the Cloud Manager Connector will be created.</td>
</tr>
<tr>
<td>aws_connector_key_name</td>
<td>String</td>
<td>(Required) The name of the key pair to use for the Connector instance.</td>
</tr>
<tr>
<td>aws_connector_company</td>
<td>String</td>
<td>(Required) The name of the company of the user.</td>
</tr>
<tr>
<td>aws_connector_instance_type</td>
<td>String</td>
<td>(Required) The type of instance (for example, t3.xlarge). At least 4 CPU and 16 GB of memory are required.</td>
</tr>
<tr>
<td>aws_connector_subnet_id</td>
<td>String</td>
<td>(Required) The ID of the subnet for the instance.</td>
</tr>
<tr>
<td>aws_connector_security_group_id</td>
<td>String</td>
<td>(Required) The ID of the security group for the instance, multiple security groups can be provided separated by ','.</td>
</tr>
<tr>
<td>aws_connector_iam_instance_profile_name</td>
<td>String</td>
<td>(Required) The name of the instance profile for the Connector.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>aws_connector_account_id</td>
<td>String</td>
<td>(Optional) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
<tr>
<td>aws_connector_public_ip_bool</td>
<td>Bool</td>
<td>(Optional) Indicates whether to associate a public IP address to the instance. If not provided, the association will be done based on the subnet’s configuration.</td>
</tr>
</tbody>
</table>

**Single Node Instance**

Terraform variables for single NetApp CVO instance.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cvo_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Volumes ONTAP working environment.</td>
</tr>
<tr>
<td>cvo_region</td>
<td>String</td>
<td>(Required) The region where the working environment will be created.</td>
</tr>
<tr>
<td>cvo_subnet_id</td>
<td>String</td>
<td>(Required) The subnet id where the working environment will be created.</td>
</tr>
<tr>
<td>cvo_vpc_id</td>
<td>String</td>
<td>(Optional) The VPC ID where the working environment will be created. If this argument isn’t provided, the VPC will be calculated by using the provided subnet ID.</td>
</tr>
<tr>
<td>cvo_svm_password</td>
<td>String</td>
<td>(Required) The admin password for Cloud Volumes ONTAP.</td>
</tr>
<tr>
<td>cvo_writing_speed_state</td>
<td>String</td>
<td>(Optional) The write speed setting for Cloud Volumes ONTAP: ['NORMAL','HIGH']. The default is 'NORMAL'.</td>
</tr>
</tbody>
</table>

**CVO HA Deployment**

Terraform configuration files for deployment of NetApp CVO (HA Pair) on AWS

This section contains various Terraform configuration files to deploy/configure NetApp CVO (Cloud Volumes ONTAP) in high availability pair on AWS (Amazon Web Services).

Terraform Documentation: https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs

**Procedure**

In order to run the template:

a. Clone the repository.

   ```
   git clone https://github.com/NetApp/na_cloud_volumes_automation.git
   ```

b. Navigate to the desired folder
c. Configure AWS credentials from the CLI.

```
aws configure
```

- AWS Access Key ID [None]: accesskey
- AWS Secret Access Key [None]: secretkey
- Default region name [None]: us-west-2
- Default output format [None]: json

d. Update the variable values in `vars/aws_cvo_ha_deployment.tfvars`.

```
You can choose to deploy the connector by setting the variable "aws_connector_deploy_bool" value to true/false.
```

e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```
terraform init
```

f. Verify the terraform files using `terraform validate` command.

```
terraform validate
```

g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

```
terraform plan -target="module.aws_ha" -var -file="vars/aws_cvo_ha_deployment.tfvars"
```

h. Run the deployment

```
terraform apply -target="module.aws_ha" -var -file="vars/aws_cvo_ha_deployment.tfvars"
```

To delete the deployment

```
terraform destroy
```
Terraform variables for NetApp AWS connector instance for CVO deployment.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>aws_connector_deploy_bool</code></td>
<td>Bool</td>
<td>(Required) Check for Connector deployment.</td>
</tr>
<tr>
<td><code>aws_connector_name</code></td>
<td>String</td>
<td>(Required) The name of the Cloud Manager Connector.</td>
</tr>
<tr>
<td><code>aws_connector_region</code></td>
<td>String</td>
<td>(Required) The region where the Cloud Manager Connector will be created.</td>
</tr>
<tr>
<td><code>aws_connected_key_name</code></td>
<td>String</td>
<td>(Required) The name of the key pair to use for the Connector instance.</td>
</tr>
<tr>
<td><code>aws_connected_company</code></td>
<td>String</td>
<td>(Required) The name of the company of the user.</td>
</tr>
<tr>
<td><code>aws_connected_instance_type</code></td>
<td>String</td>
<td>(Required) The type of instance (for example, t3.xlarge). At least 4 CPU and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16 GB of memory are required.</td>
</tr>
<tr>
<td><code>aws_connected_subnet_id</code></td>
<td>String</td>
<td>(Required) The ID of the subnet for the instance.</td>
</tr>
<tr>
<td><code>aws_connected_security_group_id</code></td>
<td>String</td>
<td>(Required) The ID of the security group for the instance, multiple security</td>
</tr>
<tr>
<td></td>
<td></td>
<td>groups can be provided separated by ','.</td>
</tr>
<tr>
<td><code>aws_connected_iam_instance_profile_name</code></td>
<td>String</td>
<td>(Required) The name of the instance profile for the Connector.</td>
</tr>
<tr>
<td><code>aws_connected_account_id</code></td>
<td>String</td>
<td>(Optional) The NetApp account ID that the Connector will be associated with.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If not provided, Cloud Manager uses the first account. If no account exists,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cloud Manager creates a new account. You can find the account ID in the ac</td>
</tr>
<tr>
<td></td>
<td></td>
<td>count tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
<tr>
<td><code>aws_connected_public_ip_bool</code></td>
<td>Bool</td>
<td>(Optional) Indicates whether to associate a public IP address to the instance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If not provided, the association will be done based on the subnet’s config</td>
</tr>
</tbody>
</table>

HA Pair

Terraform variables for NetApp CVO instances in HA Pair.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cvo_is_ha</code></td>
<td>Bool</td>
<td>(Optional) Indicate whether the working environment is an HA pair or not [true, false]. The default is false.</td>
</tr>
<tr>
<td><code>cvo_name</code></td>
<td>String</td>
<td>(Required) The name of the Cloud Volumes ONTAP working environment.</td>
</tr>
<tr>
<td><code>cvo_region</code></td>
<td>String</td>
<td>(Required) The region where the working environment will be created.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>cvo_node1_subnet_id</code></td>
<td>String</td>
<td>(Required) The subnet id where the first node will be created.</td>
</tr>
<tr>
<td><code>cvo_node2_subnet_id</code></td>
<td>String</td>
<td>(Required) The subnet id where the second node will be created.</td>
</tr>
<tr>
<td><code>cvo_vpc_id</code></td>
<td>String</td>
<td>(Optional) The VPC ID where the working environment will be created.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If this argument isn’t provided, the VPC will be calculated by</td>
</tr>
<tr>
<td></td>
<td></td>
<td>using the provided subnet ID.</td>
</tr>
<tr>
<td><code>cvo_svm_password</code></td>
<td>String</td>
<td>(Required) The admin password for Cloud Volumes ONTAP.</td>
</tr>
<tr>
<td><code>cvo_failover_mode</code></td>
<td>String</td>
<td>(Optional) For HA, the failover mode for the HA pair: ['PrivateIP',</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'FloatingIP']. 'PrivateIP' is for a single availability zone and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'FloatingIP' is for multiple availability zones.</td>
</tr>
<tr>
<td><code>cvo_mediator_subnet_id</code></td>
<td>String</td>
<td>(Optional) For HA, the subnet ID of the mediator.</td>
</tr>
<tr>
<td><code>cvo_mediator_key_pair_name</code></td>
<td>String</td>
<td>(Optional) For HA, the key pair name for the mediator instance.</td>
</tr>
<tr>
<td><code>cvo_cluster_floating_ip</code></td>
<td>String</td>
<td>(Optional) For HA FloatingIP, the cluster management floating IP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>address.</td>
</tr>
<tr>
<td><code>cvo_data_floating_ip</code></td>
<td>String</td>
<td>(Optional) For HA FloatingIP, the data floating IP address.</td>
</tr>
<tr>
<td><code>cvo_data_floating_ip2</code></td>
<td>String</td>
<td>(Optional) For HA FloatingIP, the data floating IP address.</td>
</tr>
<tr>
<td><code>cvo_svm_floating_ip</code></td>
<td>String</td>
<td>(Optional) For HA FloatingIP, the SVM management floating IP address.</td>
</tr>
<tr>
<td><code>cvo_route_table_ids</code></td>
<td>List</td>
<td>(Optional) For HA FloatingIP, the list of route table IDs that will be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>updated with the floating IPs.</td>
</tr>
</tbody>
</table>

**FSx Deployment**

**Terraform configuration files for deployment of NetApp ONTAP FSx on AWS**

This section contains various Terraform configuration files to deploy/configure NetApp ONTAP FSx on AWS (Amazon Web Services).

**Terraform Documentation:** [https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs](https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs)

**Procedure**

In order to run the template:

a. Clone the repository.

```
  git clone https://github.com/NetApp/na_cloud_volumes_automation.git
```
b. Navigate to the desired folder

```
cd na_cloud_volumes_automation/
```

c. Configure AWS credentials from the CLI.

```
aws configure
```

- AWS Access Key ID [None]: accesskey
- AWS Secret Access Key [None]: secretkey
- Default region name [None]: us-west-2
- Default output format [None]:

d. Update the variable values in `vars/aws_fsx_deployment.tfvars`

You can choose to deploy the connector by setting the variable "aws_connector_deploy_bool" value to true/false.

e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```
terraform init
```

f. Verify the terraform files using terraform validate command.

```
terraform validate
```

g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

```
terraform plan -target="module.aws_fsx" -var
-file="vars/aws_fsx_deployment.tfvars"
```

h. Run the deployment

```
terraform apply -target="module.aws_fsx" -var
-file="vars/aws_fsx_deployment.tfvars"
```

To delete the deployment

```
terraform destroy
```
## Recipes:

### Connector

Terraform variables for NetApp AWS connector instance.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aws_connector_deploy_bool</td>
<td>Bool</td>
<td>(Required) Check for Connector deployment.</td>
</tr>
<tr>
<td>aws_connector_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Manager Connector.</td>
</tr>
<tr>
<td>aws_connector_region</td>
<td>String</td>
<td>(Required) The region where the Cloud Manager Connector will be created.</td>
</tr>
<tr>
<td>aws_connector_key_name</td>
<td>String</td>
<td>(Required) The name of the key pair to use for the Connector instance.</td>
</tr>
<tr>
<td>aws_connector_company</td>
<td>String</td>
<td>(Required) The name of the company of the user.</td>
</tr>
<tr>
<td>aws_connector_instance_type</td>
<td>String</td>
<td>(Required) The type of instance (for example, t3.xlarge). At least 4 CPU and 16 GB of memory are required.</td>
</tr>
<tr>
<td>aws.connector_subnet_id</td>
<td>String</td>
<td>(Required) The ID of the subnet for the instance.</td>
</tr>
<tr>
<td>aws.connector_security_group_id</td>
<td>String</td>
<td>(Required) The ID of the security group for the instance, multiple security groups can be provided separated by ','.</td>
</tr>
<tr>
<td>aws.connector_instance_profile_name</td>
<td>String</td>
<td>(Required) The name of the instance profile for the Connector.</td>
</tr>
<tr>
<td>aws.connector_account_id</td>
<td>String</td>
<td>(Optional) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
<tr>
<td>aws.connector_public_ip_bool</td>
<td>Bool</td>
<td>(Optional) Indicates whether to associate a public IP address to the instance. If not provided, the association will be done based on the subnet's configuration.</td>
</tr>
</tbody>
</table>

### FSx Instance

Terraform variables for NetApp ONTAP FSx instance.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fsx_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Volumes ONTAP working environment.</td>
</tr>
<tr>
<td>fsx_region</td>
<td>String</td>
<td>(Required) The region where the working environment will be created.</td>
</tr>
<tr>
<td>fsx_primary_subnet_id</td>
<td>String</td>
<td>(Required) The primary subnet id where the working environment will be created.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>fsx_secondary_subnet_id</td>
<td>String</td>
<td>(Required) The secondary subnet id where the working environment will be created.</td>
</tr>
<tr>
<td>fsx_account_id</td>
<td>String</td>
<td>(Required) The NetApp account ID that the FSx instance will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
<tr>
<td>fsx_workspace_id</td>
<td>String</td>
<td>(Required) The ID of the Cloud Manager workspace of working environment.</td>
</tr>
<tr>
<td>fsx_admin_password</td>
<td>String</td>
<td>(Required) The admin password for Cloud Volumes ONTAP.</td>
</tr>
<tr>
<td>fsx_throughput_capacity</td>
<td>String</td>
<td>(Optional) capacity of the throughput.</td>
</tr>
<tr>
<td>fsx_storage_capacity_size</td>
<td>String</td>
<td>(Optional) EBS volume size for the first data aggregate. For GB, the unit can be: [100 or 500]. For TB, the unit can be: [1,2,4,8,16]. The default is '1'</td>
</tr>
<tr>
<td>fsx_storage_capacity_size_unit</td>
<td>String</td>
<td>(Optional) ['GB' or 'TB']. The default is 'TB'.</td>
</tr>
<tr>
<td>fsx_cloudmanager_aws_credentials_name</td>
<td>String</td>
<td>(Required) The name of the AWS Credentials account name.</td>
</tr>
</tbody>
</table>
Azure
ANF
Terraform configuration files for deployment of ANF Volume on Azure
This section contains various Terraform configuration files to deploy/configure ANF (Azure Netapp Files) Volume on Azure.

Terraform Documentation: https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs

Procedure
In order to run the template:

a. Clone the repository.

```bash
git clone https://github.com/NetApp/na_cloud_volumes_automation.git
```

b. Navigate to the desired folder

```bash
cd na_cloud_volumes_automation
```

c. Login to your Azure CLI (Azure CLI must be installed).

```bash
az login
```

d. Update the variable values in `vars/azure_anf.tfvars`.

You can choose to deploy the ANF volume using an existing vnet and subnet by setting the variable "vnet_creation_bool" and "subnet_creation_bool" value to false and supplying the "subnet_id_for_anf_vol". You can also set those values to true and create a new vnet and subnet in which case, the subnet ID will automatically be taken from the newly created subnet.

e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```bash
terraform init
```

f. Verify the terraform files using terraform validate command.

```bash
terraform validate
```

g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.
terraform plan -target="module.anf" -var-file="vars/azure_anf.tfvars"

h. Run the deployment

terraform apply -target="module.anf" -var-file="vars/azure_anf.tfvars"

To delete the deployment

terraform destroy

Recipes:
Single Node Instance

Terraform variables for single NetApp ANF Volume.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>az_location</td>
<td>String</td>
<td>(Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_prefix</td>
<td>String</td>
<td>(Required) The name of the resource group where the NetApp Volume should be created. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_vnet_address_space</td>
<td>String</td>
<td>(Required) The address space to be used by the newly created vnet for ANF volume deployment.</td>
</tr>
<tr>
<td>az_subnet_address_prefix</td>
<td>String</td>
<td>(Required) The subnet address prefix to be used by the newly created vnet for ANF volume deployment.</td>
</tr>
<tr>
<td>az_volume_path</td>
<td>String</td>
<td>(Required) A unique file path for the volume. Used when creating mount targets. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_capacity_pool_size</td>
<td>Integer</td>
<td>(Required) Capacity Pool Size mentioned in TB.</td>
</tr>
<tr>
<td>az_vnet_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true if you want to create a new vnet. Set it to false to use an existing vnet.</td>
</tr>
<tr>
<td>az_subnet_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true to create a new subnet. Set it to false to use an existing subnet.</td>
</tr>
<tr>
<td>az_subnet_id_for_anf_vol</td>
<td>String</td>
<td>(Required) Mention the subnet id in case you decide to use an existing subnet by setting subnet_creation_bool to true. If set to false, leave it at the default value.</td>
</tr>
<tr>
<td>az_netapp_pool_service_level</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_netapp_vol_service_level</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_vol_protocol</td>
<td>String</td>
<td>(Optional) The target volume protocol expressed as a list. Supported single value include CIFS, NFSv3, or NFSv4.1. If argument is not defined it will default to NFSv3. Changing this forces a new resource to be created and data will be lost.</td>
</tr>
<tr>
<td>az_netapp_vol_security_style</td>
<td>String</td>
<td>(Optional) Volume security style, accepted values are Unix or Ntfs. If not provided, single-protocol volume is created defaulting to Unix if it is NFSv3 or NFSv4.1 volume, if CIFS, it will default to Ntfs. In a dual-protocol volume, if not provided, its value will be Ntfs.</td>
</tr>
<tr>
<td>az_netapp_vol_storage_quota</td>
<td>String</td>
<td>(Required) The maximum Storage Quota allowed for a file system in Gigabytes.</td>
</tr>
</tbody>
</table>

As per the recommendation, this script uses the `prevent_destroy` lifecycle argument to mitigate the possibility of accidental data loss in the configuration file. For more information on the `prevent_destroy` lifecycle argument please see the terraform documentation: https://developer.hashicorp.com/terraform/tutorials/state/resource-lifecycle#prevent-resource-deletion.

**ANF Data Protection**

**Terraform configuration files for deployment of ANF Volume with Data Protection on Azure**

This section contains various Terraform configuration files to deploy/configure ANF (Azure Netapp Files) Volume with Data Protection on Azure.

Terraform Documentation: https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs

**Procedure**

In order to run the template:

1. Clone the repository.
   
   ```bash
   git clone https://github.com/NetApp/na_cloud_volumes_automation.git
   ```

2. Navigate to the desired folder
   
   ```bash
   cd na_cloud_volumes_automation
   ```

3. Login to your Azure CLI (Azure CLI must be installed).
   
   ```bash
   az login
   ```

4. Update the variable values in `vars/azure_anf_data_protection.tfvars`.
You can choose to deploy the ANF volume using an existing vnet and subnet by setting the variable "vnet_creation_bool" and "subnet_creation_bool" value to false and supplying the "subnet_id_for_anf_vol". You can also set those values to true and create a new vnet and subnet in which case, the subnet ID will automatically be taken from the newly created subnet.

e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

   `terraform init`

f. Verify the terraform files using terraform validate command.

   `terraform validate`

g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

   `terraform plan -target="module.anf_data_protection" -var -file="vars/azure_anf_data_protection.tfvars"`

h. Run the deployment

   `terraform apply -target="module.anf_data_protection" -var -file="vars/azure_anf_data_protection.tfvars`

To delete the deployment

   `terraform destroy`

**Recipes:**

**ANF Data Protection**

Terraform variables for single ANF Volume with data protection enabled.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>az_location</td>
<td>String</td>
<td>(Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_alt_location</td>
<td>String</td>
<td>(Required) The Azure location where the secondary volume will be created.</td>
</tr>
<tr>
<td>az_prefix</td>
<td>String</td>
<td>(Required) The name of the resource group where the NetApp Volume should be created. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_vnet_primary_address_space</td>
<td>String</td>
<td>(Required) The address space to be used by the newly created vnet for ANF primary volume deployment.</td>
</tr>
<tr>
<td>az_vnet_secondary_address_space</td>
<td>String</td>
<td>(Required) The address space to be used by the newly created vnet for ANF secondary volume deployment.</td>
</tr>
<tr>
<td>az_subnet_primary_address_prefix</td>
<td>String</td>
<td>(Required) The subnet address prefix to be used by the newly created vnet for ANF primary volume deployment.</td>
</tr>
<tr>
<td>az_subnet_secondary_address_prefix</td>
<td>String</td>
<td>(Required) The subnet address prefix to be used by the newly created vnet for ANF secondary volume deployment.</td>
</tr>
<tr>
<td>az_volume_path_primary</td>
<td>String</td>
<td>(Required) A unique file path for the primary volume. Used when creating mount targets. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_volume_path_secondary</td>
<td>String</td>
<td>(Required) A unique file path for the secondary volume. Used when creating mount targets. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_capacity_pool_size_primary</td>
<td>Integer</td>
<td>(Required) Capacity Pool Size mentioned in TB.</td>
</tr>
<tr>
<td>az_capacity_pool_size_secondary</td>
<td>Integer</td>
<td>(Required) Capacity Pool Size mentioned in TB.</td>
</tr>
<tr>
<td>az_vnet_primary_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true if you want to create a new vnet for primary volume. Set it to false to use an existing vnet.</td>
</tr>
<tr>
<td>az_vnet_secondary_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true if you want to create a new vnet for secondary volume. Set it to false to use an existing vnet.</td>
</tr>
<tr>
<td>az_subnet_primary_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true to create a new subnet for primary volume. Set it to false to use an existing subnet.</td>
</tr>
<tr>
<td>az_subnet_secondary_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true to create a new subnet for secondary volume. Set it to false to use an existing subnet.</td>
</tr>
<tr>
<td>az_primary_subnet_id_for_anf_vol</td>
<td>String</td>
<td>(Required) Mention the subnet id in case you decide to use an existing subnet by setting subnet_primary_creation_bool to true. If set to false, leave it at the default value.</td>
</tr>
<tr>
<td>az_secondary_subnet_id_for_anf_vol</td>
<td>String</td>
<td>(Required) Mention the subnet id in case you decide to use an existing subnet by setting subnet_secondary_creation_bool to true. If set to false, leave it at the default value.</td>
</tr>
<tr>
<td>az_netapp_pool_service_level_primary</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_pool_service_level_secondary</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_netapp_vol_service_level_primary</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_vol_service_level_secondary</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_vol_protocol_primary</td>
<td>String</td>
<td>(Optional) The target volume protocol expressed as a list. Supported single value include CIFS, NFSv3, or NFSv4.1. If argument is not defined it will default to NFSv3. Changing this forces a new resource to be created and data will be lost.</td>
</tr>
<tr>
<td>az_netapp_vol_protocol_secondary</td>
<td>String</td>
<td>(Optional) The target volume protocol expressed as a list. Supported single value include CIFS, NFSv3, or NFSv4.1. If argument is not defined it will default to NFSv3. Changing this forces a new resource to be created and data will be lost.</td>
</tr>
<tr>
<td>az_netapp_vol_storage_quota_primary</td>
<td>String</td>
<td>(Required) The maximum Storage Quota allowed for a file system in Gigabytes.</td>
</tr>
<tr>
<td>az_netapp_vol_storage_quota_secondary</td>
<td>String</td>
<td>(Required) The maximum Storage Quota allowed for a file system in Gigabytes.</td>
</tr>
<tr>
<td>az_dp_repllication_frequency</td>
<td>String</td>
<td>(Required) Replication frequency, supported values are 10minutes, hourly, daily, values are case sensitive.</td>
</tr>
</tbody>
</table>

As per the recommendation, this script uses the `prevent_destroy` lifecycle argument to mitigate the possibility of accidental data loss in the configuration file. For more information on the `prevent_destroy` lifecycle argument please see the terraform documentation: https://developer.hashicorp.com/terraform/tutorials/state/resource-lifecycle#prevent-resource-deletion.

ANF Dual Protocol
Terraform configuration files for deployment of ANF Volume with dual protocol on Azure
This section contains various Terraform configuration files to deploy/configure ANF (Azure Netapp Files) Volume with dual protocol enabled on Azure.

Terraform Documentation: https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs

Procedure
In order to run the template:

a. Clone the repository.

```bash
git clone https://github.com/NetApp/na_cloud_volumes_automation.git
```

b. Navigate to the desired folder
cd na_cloud_volumes_automation

c. Login to your Azure CLI (Azure CLI must be installed).

az login

d. Update the variable values in vars/azure_anf_dual_protocol.tfvars.

You can choose to deploy the ANF volume using an existing vnet and subnet by setting the variable "vnet_creation_bool" and "subnet_creation_bool" value to false and supplying the "subnet_id_for_anf_vol". You can also set those values to true and create a new vnet and subnet in which case, the subnet ID will automatically be taken from the newly created subnet.

e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

terraform init

f. Verify the terraform files using terraform validate command.

terraform validate

g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

terraform plan -target="module.anf_dual_protocol" -var -file="vars/azure_anf_dual_protocol.tfvars"

h. Run the deployment

terraform apply -target="module.anf_dual_protocol" -var -file="vars/azure_anf_dual_protocol.tfvars"

To delete the deployment

terraform destroy

Recipes:
Single Node Instance
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>az_location</td>
<td>String</td>
<td>(Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_prefix</td>
<td>String</td>
<td>(Required) The name of the resource group where the NetApp Volume should be created. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_vnet_address_space</td>
<td>String</td>
<td>(Required) The address space to be used by the newly created vnet for ANF volume deployment.</td>
</tr>
<tr>
<td>az_subnet_address_prefix</td>
<td>String</td>
<td>(Required) The subnet address prefix to be used by the newly created vnet for ANF volume deployment.</td>
</tr>
<tr>
<td>az_volume_path</td>
<td>String</td>
<td>(Required) A unique file path for the volume. Used when creating mount targets. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_capacity_pool_size</td>
<td>Integer</td>
<td>(Required) Capacity Pool Size mentioned in TB.</td>
</tr>
<tr>
<td>az_vnet_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true if you want to create a new vnet. Set it to false to use an existing vnet.</td>
</tr>
<tr>
<td>az_subnet_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true to create a new subnet. Set it to false to use an existing subnet.</td>
</tr>
<tr>
<td>az_subnet_id_for_anf_vol</td>
<td>String</td>
<td>(Required) Mention the subnet id in case you decide to use an existing subnet by setting subnet_creation_bool to true. If set to false, leave it at the default value.</td>
</tr>
<tr>
<td>az_netapp_pool_service_level</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_vol_service_level</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_vol_protocol1</td>
<td>String</td>
<td>(Required) The target volume protocol expressed as a list. Supported single value include CIFS, NFSv3, or NFSv4.1. If argument is not defined it will default to NFSv3. Changing this forces a new resource to be created and data will be lost.</td>
</tr>
<tr>
<td>az_netapp_vol_protocol2</td>
<td>String</td>
<td>(Required) The target volume protocol expressed as a list. Supported single value include CIFS, NFSv3, or NFSv4.1. If argument is not defined it will default to NFSv3. Changing this forces a new resource to be created and data will be lost.</td>
</tr>
<tr>
<td>az_netapp_vol_storage_quota</td>
<td>String</td>
<td>(Required) The maximum Storage Quota allowed for a file system in Gigabytes.</td>
</tr>
<tr>
<td>az_smb_server_username</td>
<td>String</td>
<td>(Required) Username to create ActiveDirectory object.</td>
</tr>
<tr>
<td>az_smb_server_password</td>
<td>String</td>
<td>(Required) User Password to create ActiveDirectory object.</td>
</tr>
<tr>
<td>az_smb_server_name</td>
<td>String</td>
<td>(Required) Server Name to create ActiveDirectory object.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_smb_dns_servers</td>
<td>String</td>
<td>(Required) DNS Server IP to create ActiveDirectory object.</td>
</tr>
</tbody>
</table>

As per the recommendation, this script uses the `prevent_destroy` lifecycle argument to mitigate the possibility of accidental data loss in the configuration file. For more information on the `prevent_destroy` lifecycle argument please see the Terraform documentation: https://developer.hashicorp.com/terraform/tutorials/state/resource-lifecycle#prevent-resource-deletion.

**ANF Volume From Snapshot**

**Terraform configuration files for deployment of ANF Volume from Snapshot on Azure**

This section contains various Terraform configuration files to deploy/configure ANF (Azure Netapp Files) Volume from Snapshot on Azure.

**Terraform Documentation:** https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs

**Procedure**

In order to run the template:

1. Clone the repository.
   
go to https://github.com/NetApp/na_cloud_volumes_automation.git

2. Navigate to the desired folder
   
   cd na_cloud_volumes_automation

3. Login to your Azure CLI (Azure CLI must be installed).
   
   az login

4. Update the variable values in `vars/azure_anf_volume_from_snapshot.tfvars`.

   You can choose to deploy the ANF volume using an existing vnet and subnet by setting the variable "vnet_creation_bool" and "subnet_creation_bool" value to false and supplying the "subnet_id_for_anf_vol". You can also set those values to true and create a new vnet and subnet in which case, the subnet ID will automatically be taken from the newly created subnet.

5. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.
   
   terraform init
b. Verify the terraform files using terraform validate command.

```
terraform validate
```

c. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

```
terraform plan -target="module.anf_volume_from_snapshot" -var-file="vars/azure_anf_volume_from_snapshot.tfvars"
```

d. Run the deployment

```
terraform apply -target="module.anf_volume_from_snapshot" -var-file="vars/azure_anf_volume_from_snapshot.tfvars"
```

To delete the deployment

```
terraform destroy
```

Recipies:

Single Node Instance

Terraform variables for single ANF Volume using snapshot.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>az_location</td>
<td>String</td>
<td>(Required) Specifies the supported Azure location where the resource exists. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_prefix</td>
<td>String</td>
<td>(Required) The name of the resource group where the NetApp Volume should be created. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_vnet_address_space</td>
<td>String</td>
<td>(Required) The address space to be used by the newly created vnet for ANF volume deployment.</td>
</tr>
<tr>
<td>az_subnet_address_prefix</td>
<td>String</td>
<td>(Required) The subnet address prefix to be used by the newly created vnet for ANF volume deployment.</td>
</tr>
<tr>
<td>az_volume_path</td>
<td>String</td>
<td>(Required) A unique file path for the volume. Used when creating mount targets. Changing this forces a new resource to be created.</td>
</tr>
<tr>
<td>az_capacity_pool_size</td>
<td>Integer</td>
<td>(Required) Capacity Pool Size mentioned in TB.</td>
</tr>
<tr>
<td>az_vnet_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true if you want to create a new vnet. Set it to false to use an existing vnet.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>----------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_subnet_creation_bool</td>
<td>Boolean</td>
<td>(Required) Set this boolean to true to create a new subnet. Set it to false to use an existing subnet.</td>
</tr>
<tr>
<td>az_subnet_id_for_anf_vol</td>
<td>String</td>
<td>(Required) Mention the subnet id in case you decide to use an existing subnet by setting subnet_creation_bool to true. If set to false, leave it at the default value.</td>
</tr>
<tr>
<td>az_netapp_pool_service_level</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_vol_service_level</td>
<td>String</td>
<td>(Required) The target performance of the file system. Valid values include Premium, Standard, or Ultra.</td>
</tr>
<tr>
<td>az_netapp_vol_protocol</td>
<td>String</td>
<td>(Optional) The target volume protocol expressed as a list. Supported single value include CIFS, NFSv3, or NFSv4.1. If argument is not defined it will default to NFSv3. Changing this forces a new resource to be created and data will be lost.</td>
</tr>
<tr>
<td>az_netapp_vol_storage_quota</td>
<td>String</td>
<td>(Required) The maximum Storage Quota allowed for a file system in Gigabytes.</td>
</tr>
<tr>
<td>az_snapshot_id</td>
<td>String</td>
<td>(Required) Snapshot ID using which new ANF volume will be created.</td>
</tr>
</tbody>
</table>

As per the recommendation, this script uses the `prevent_destroy` lifecycle argument to mitigate the possibility of accidental data loss in the configuration file. For more information on the `prevent_destroy` lifecycle argument please see the terraform documentation: https://developer.hashicorp.com/terraform/tutorials/state/resource-lifecycle#prevent-resource-deletion.

CVO Single Node Deployment

Terraform configuration files for deployment of Single Node CVO on Azure

This section contains various Terraform configuration files to deploy/configure Single Node CVO (Cloud Volumes ONTAP) on Azure.

Terraform Documentation: https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs

Procedure

In order to run the template:

a. Clone the repository.

```bash
git clone https://github.com/NetApp/na_cloud_volumes_automation.git
```

b. Navigate to the desired folder

```bash
cd na_cloud_volumes_automation
```
c. Login to your Azure CLI (Azure CLI must be installed).

```bash
az login
```

d. Update the variables in `vars\azure_cvo_single_node_deployment.tfvars`.

e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```bash
terraform init
```

f. Verify the Terraform files using `terraform validate` command.

```bash
terraform validate
```

g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

```bash
terraform plan
  -target="module.az_cvo_single_node_deployment" -var
  -file="vars\azure_cvo_single_node_deployment.tfvars"
```

h. Run the deployment

```bash
terraform apply
  -target="module.az_cvo_single_node_deployment" -var
  -file="vars\azure_cvo_single_node_deployment.tfvars"
```

To delete the deployment

```bash
terraform destroy
```

**Recipes:**

**Single Node Instance**

Terraform variables for single node Cloud Volumes ONTAP (CVO).

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>refresh_token</td>
<td>String</td>
<td>(Required) The refresh token of NetApp cloud manager. This can be generated from netapp Cloud Central.</td>
</tr>
<tr>
<td>az_connector_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Manager Connector.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_connector_location</td>
<td>String</td>
<td>(Required) The location where the Cloud Manager Connector will be created.</td>
</tr>
<tr>
<td>az_connector_subscription_id</td>
<td>String</td>
<td>(Required) The ID of the Azure subscription.</td>
</tr>
<tr>
<td>az_connector_company</td>
<td>String</td>
<td>(Required) The name of the company of the user.</td>
</tr>
<tr>
<td>az_connector_resource_group</td>
<td>Integer</td>
<td>(Required) The resource group in Azure where the resources will be created.</td>
</tr>
<tr>
<td>az_connector_subnet_id</td>
<td>String</td>
<td>(Required) The name of the subnet for the virtual machine.</td>
</tr>
<tr>
<td>az_connector_vnet_id</td>
<td>String</td>
<td>(Required) The name of the virtual network.</td>
</tr>
<tr>
<td>az_network_security_group_name</td>
<td>String</td>
<td>(Required) The name of the security group for the instance.</td>
</tr>
<tr>
<td>az_connecte_associate_public_ip_address</td>
<td>String</td>
<td>(Required) Indicates whether to associate the public IP address to the virtual machine.</td>
</tr>
<tr>
<td>az_connector_account_id</td>
<td>String</td>
<td>(Required) The NetApp account ID that the Connector will be associated with.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
<tr>
<td>az_connector_admin_password</td>
<td>String</td>
<td>(Required) The password for the Connector.</td>
</tr>
<tr>
<td>az_connector_admin_username</td>
<td>String</td>
<td>(Required) The user name for the Connector.</td>
</tr>
<tr>
<td>az_cvo_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Volumes ONTAP working environment.</td>
</tr>
<tr>
<td>az_cvo_location</td>
<td>String</td>
<td>(Required) The location where the working environment will be created.</td>
</tr>
<tr>
<td>az_cvo_subnet_id</td>
<td>String</td>
<td>(Required) The name of the subnet for the Cloud Volumes ONTAP system.</td>
</tr>
<tr>
<td>az_cvo_vnet_id</td>
<td>String</td>
<td>(Required) The name of the virtual network.</td>
</tr>
<tr>
<td>az_cvo_vnet_resource_group</td>
<td>String</td>
<td>(Required) The resource group in Azure associated to the virtual network.</td>
</tr>
<tr>
<td>az_cvo_data_encryption_type</td>
<td>String</td>
<td>(Required) The type of encryption to use for the working environment: [AZURE, NONE]. The default is AZURE.</td>
</tr>
<tr>
<td>az_cvo_storage_type</td>
<td>String</td>
<td>(Required) The type of storage for the first data aggregate: [Premium_LRS, Standard_LRS, StandardSSD_LRS]. The default is Premium_LRS</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_cvo_svm_password</td>
<td>String</td>
<td>(Required) The admin password for Cloud Volumes ONTAP.</td>
</tr>
<tr>
<td>az_cvo_workspace_id</td>
<td>String</td>
<td>(Required) The ID of the Cloud Manager workspace where you want to deploy Cloud Volumes ONTAP. If not provided, Cloud Manager uses the first workspace. You can find the ID from the Workspace tab on <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
<tr>
<td>az_cvo_capacity_tier</td>
<td>String</td>
<td>(Required) Whether to enable data tiering for the first data aggregate: [Blob, NONE]. The default is BLOB.</td>
</tr>
<tr>
<td>az_cvo_writing_speed_state</td>
<td>String</td>
<td>(Required) The write speed setting for Cloud Volumes ONTAP: [NORMAL, HIGH]. The default is NORMAL. This argument is not relevant for HA pairs.</td>
</tr>
<tr>
<td>az_cvo_ontap_version</td>
<td>String</td>
<td>(Required) The required ONTAP version. Ignored if 'use_latest_version' is set to true. The default is to use the latest version.</td>
</tr>
<tr>
<td>az_cvo_instance_type</td>
<td>String</td>
<td>(Required) The type of instance to use, which depends on the license type you chose: Explore: [Standard_DS3_v2], Standard: [Standard_DS4_v2, Standard_DS13_v2, Standard_L8s_v2], Premium: [Standard_DS5_v2, Standard_DS14_v2], BYOL: all instance types defined for PayGo. For more supported instance types, refer to Cloud Volumes ONTAP Release Notes. The default is Standard_DS4_v2.</td>
</tr>
<tr>
<td>az_cvo_license_type</td>
<td>String</td>
<td>(Required) The type of license to be use. For single node: [azure-cot-explore-paygo, azure-cot-standard-paygo, azure-cot-premium-paygo, azure-cot-premium-byol, capacity-paygo]. For HA: [azure-ha-cot-standard-paygo, azure-ha-cot-premium-paygo, azure-ha-cot-premium-byol, ha-capacity-paygo]. The default is azure-cot-standard-paygo. Use capacity-paygo or ha-capacity-paygo for HA on selecting Bring Your Own License type Capacity-Based or Freemium. Use azure-cot-premium-byol or azure-ha-cot-premium-byol for HA on selecting Bring Your Own License type Node-Based.</td>
</tr>
<tr>
<td>az_cvo_nss_account</td>
<td>String</td>
<td>(Required) The NetApp Support Site account ID to use with this Cloud Volumes ONTAP system. If the license type is BYOL and an NSS account isn’t provided, Cloud Manager tries to use the first existing NSS account.</td>
</tr>
<tr>
<td>az_tenant_id</td>
<td>String</td>
<td>(Required) Tenant ID of the application/service principal registered in Azure.</td>
</tr>
<tr>
<td>az_application_id</td>
<td>String</td>
<td>(Required) Application ID of the application/service principal registered in Azure.</td>
</tr>
<tr>
<td>az_application_key</td>
<td>String</td>
<td>(Required) The Application Key of the application/service principal registered in Azure.</td>
</tr>
</tbody>
</table>

**CVO HA Deployment**

**Terraform configuration files for deployment of CVO HA on Azure**

This section contains various Terraform configuration files to deploy/configure CVO (Cloud Volumes
ONTAP) HA (High Availability) on Azure.


**Procedure**

In order to run the template:

a. Clone the repository.

```bash
  git clone https://github.com/NetApp/na_cloud_volumes_automation.git
```

b. Navigate to the desired folder

```bash
  cd na_cloud_volumes_automation
```

c. Login to your Azure CLI (Azure CLI must be installed).

```bash
  az login
```

d. Update the variables in `vars\azure_cvo_ha_deployment.tfvars`.

e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```bash
  terraform init
```

f. Verify the terraform files using `terraform validate` command.

```bash
  terraform validate
```

g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

```bash
  terraform plan -target="module.az_cvo_ha_deployment" -var -file="vars\azure_cvo_ha_deployment.tfvars"
```

h. Run the deployment

```bash
  terraform apply -target="module.az_cvo_ha_deployment" -var -file="vars\azure_cvo_ha_deployment.tfvars"
```
To delete the deployment

```bash
terraform destroy
```

Recipes:

**HA Pair Instance**

Terraform variables for HA pair Cloud Volumes ONTAP (CVO).

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>refresh_token</td>
<td>String</td>
<td>(Required) The refresh token of NetApp cloud manager. This can be generated from netapp Cloud Central.</td>
</tr>
<tr>
<td>az_connector_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Manager Connector.</td>
</tr>
<tr>
<td>az_connector_location</td>
<td>String</td>
<td>(Required) The location where the Cloud Manager Connector will be created.</td>
</tr>
<tr>
<td>az_connector_subscription_id</td>
<td>String</td>
<td>(Required) The ID of the Azure subscription.</td>
</tr>
<tr>
<td>az_connector_company</td>
<td>String</td>
<td>(Required) The name of the company of the user.</td>
</tr>
<tr>
<td>az_connector_resource_group</td>
<td>Integer</td>
<td>(Required) The resource group in Azure where the resources will be created.</td>
</tr>
<tr>
<td>az_connector_subnet_id</td>
<td>String</td>
<td>(Required) The name of the subnet for the virtual machine.</td>
</tr>
<tr>
<td>az_connector_vnet_id</td>
<td>String</td>
<td>(Required) The name of the virtual network.</td>
</tr>
<tr>
<td>az_connector_network_security_group_name</td>
<td>String</td>
<td>(Required) The name of the security group for the instance.</td>
</tr>
<tr>
<td>az_connector_associate_public_ip_address</td>
<td>String</td>
<td>(Required) Indicates whether to associate the public IP address to the virtual machine.</td>
</tr>
<tr>
<td>az_connector_account_id</td>
<td>String</td>
<td>(Required) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
<tr>
<td>az_connector_admin_password</td>
<td>String</td>
<td>(Required) The password for the Connector.</td>
</tr>
<tr>
<td>az_connector_admin_username</td>
<td>String</td>
<td>(Required) The user name for the Connector.</td>
</tr>
<tr>
<td>az_cvo_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Volumes ONTAP working environment.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>az_cvo_location</td>
<td>String</td>
<td>(Required) The location where the working environment will be created.</td>
</tr>
<tr>
<td>az_cvo_subnet_id</td>
<td>String</td>
<td>(Required) The name of the subnet for the Cloud Volumes ONTAP system.</td>
</tr>
<tr>
<td>az_cvo_vnet_id</td>
<td>String</td>
<td>(Required) The name of the virtual network.</td>
</tr>
<tr>
<td>az_cvo_vnet_resource_group</td>
<td>String</td>
<td>(Required) The resource group in Azure associated to the virtual network.</td>
</tr>
<tr>
<td>az_cvo_data_encryption_type</td>
<td>String</td>
<td>(Required) The type of encryption to use for the working environment: [AZURE, NONE]. The default is AZURE.</td>
</tr>
<tr>
<td>az_cvo_storage_type</td>
<td>String</td>
<td>(Required) The type of storage for the first data aggregate: [Premium_LRS, Standard_LRS, StandardSSD_LRS]. The default is Premium_LRS.</td>
</tr>
<tr>
<td>az_cvo_svm_password</td>
<td>String</td>
<td>(Required) The admin password for Cloud Volumes ONTAP.</td>
</tr>
<tr>
<td>az_cvo_workspace_id</td>
<td>String</td>
<td>(Required) The ID of the Cloud Manager workspace where you want to deploy Cloud Volumes ONTAP. If not provided, Cloud Manager uses the first workspace. You can find the ID from the Workspace tab on <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
<tr>
<td>az_cvo_capacity_tier</td>
<td>String</td>
<td>(Required) Whether to enable data tiering for the first data aggregate: [Blob, NONE]. The default is BLOB.</td>
</tr>
<tr>
<td>az_cvo_writing_speed_state</td>
<td>String</td>
<td>(Required) The write speed setting for Cloud Volumes ONTAP: [NORMAL, HIGH]. The default is NORMAL. This argument is not relevant for HA pairs.</td>
</tr>
<tr>
<td>az_cvo_ontap_version</td>
<td>String</td>
<td>(Required) The required ONTAP version. Ignored if 'use_latest_version' is set to true. The default is to use the latest version.</td>
</tr>
<tr>
<td>az_cvo_instance_type</td>
<td>String</td>
<td>(Required) The type of instance to use, which depends on the license type you chose: Explore:[Standard_DS3_v2], Standard:[Standard_DS4_v2, Standard_DS13_v2, Standard_L8s_v2], Premium:[Standard_DS5_v2, Standard_DS14_v2], BYOL: all instance types defined for PayGo. For more supported instance types, refer to Cloud Volumes ONTAP Release Notes. The default is Standard_DS4_v2.</td>
</tr>
<tr>
<td>az_cvo_license_type</td>
<td>String</td>
<td>(Required) The type of license to be use. For single node: [azure-cot-explore-paygo, azure-cot-standard-paygo, azure-cot-premium-paygo, azure-cot-premium-byol, capacity-paygo]. For HA: [azure-ha-cot-standard-paygo, azure-ha-cot-premium-paygo, azure-ha-cot-premium-byol, ha-capacity-paygo]. The default is azure-cot-standard-paygo. Use capacity-paygo or ha-capacity-paygo for HA on selecting Bring Your Own License type Capacity-Based or Freemium. Use azure-cot-premium-byol or azure-ha-cot-premium-byol for HA on selecting Bring Your Own License type Node-Based.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>az_cvo_nss_account</code></td>
<td>String</td>
<td>(Required) The NetApp Support Site account ID to use with this Cloud Volumes ONTAP system. If the license type is BYOL and an NSS account isn't provided, Cloud Manager tries to use the first existing NSS account.</td>
</tr>
<tr>
<td><code>az_tenant_id</code></td>
<td>String</td>
<td>(Required) Tenant ID of the application/service principal registered in Azure.</td>
</tr>
<tr>
<td><code>az_application_id</code></td>
<td>String</td>
<td>(Required) Application ID of the application/service principal registered in Azure.</td>
</tr>
<tr>
<td><code>az_application_key</code></td>
<td>String</td>
<td>(Required) The Application Key of the application/service principal registered in Azure.</td>
</tr>
</tbody>
</table>
CVO Single Node Deployment
Terraform configuration files for deployment of NetApp CVO (Single Node Instance) on GCP
This section contains various Terraform configuration files to deploy/configure single node NetApp
CVO (Cloud Volumes ONTAP) on GCP (Google Cloud Platform).

Terraform Documentation: https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs

Procedure
In order to run the template:

a. Clone the repository.

   git clone https://github.com/NetApp/na_cloud_volumes_automation.git

b. Navigate to the desired folder

   cd na_cloud_volumes_automation/

c. Save the GCP authentication key JSON file in the directory.

d. Update the variable values in vars/gcp_cvo_single_node_deployment.tfvars

   You can choose to deploy the connector by setting the variable
   "gcp_connector_deploy_bool" value to true/false.

e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

   terraform init

f. Verify the terraform files using terraform validate command.

   terraform validate

g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

   terraform plan -target="module.gco_single_node" -var -file="vars/gcp_cvo_single_node_deployment.tfvars"

h. Run the deployment
terraform apply -target="module.gcp_single_node" -var -file="vars/gcp_cvo_single_node_deployment.tfvars"

To delete the deployment

terraform destroy

Recipies:
Connector

Terraform variables for NetApp GCP connector instance for CVO deployment.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcp_connector_deploy_bool</td>
<td>Bool</td>
<td>(Required) Check for Connector deployment.</td>
</tr>
<tr>
<td>gcp_connector_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Manager Connector.</td>
</tr>
<tr>
<td>gcp_connector_project_id</td>
<td>String</td>
<td>(Required) The GCP project_id where the connector will be created.</td>
</tr>
<tr>
<td>gcp_connector_zone</td>
<td>String</td>
<td>(Required) The GCP zone where the Connector will be created.</td>
</tr>
<tr>
<td>gcp_connector_company</td>
<td>String</td>
<td>(Required) The name of the company of the user.</td>
</tr>
<tr>
<td>gcp_connector_service_account_email</td>
<td>String</td>
<td>(Required) The email of the service_account for the connector instance. This service account is used to allow the Connector to create Cloud Volume ONTAP.</td>
</tr>
<tr>
<td>gcp_connector_service_account_path</td>
<td>String</td>
<td>(Required) The local path of the service_account JSON file for GCP authorization purposes. This service account is used to create the Connector in GCP.</td>
</tr>
<tr>
<td>gcp_connector_account_id</td>
<td>String</td>
<td>(Optional) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
</tbody>
</table>

Single Node Instance

Terraform variables for single NetApp CVO instance on GCP.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcp_cvo_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Volumes ONTAP working environment.</td>
</tr>
</tbody>
</table>
### CVO HA Deployment

**Terraform configuration files for deployment of NetApp CVO (HA Pair) on GCP**

This section contains various Terraform configuration files to deploy/configure NetApp CVO (Cloud Volumes ONTAP) in high availability pair on GCP (Google Cloud Platform).


**Procedure**

In order to run the template:

1. Clone the repository.

   ```bash
   git clone https://github.com/NetApp/na_cloud_volumes_automation.git
   ```

2. Navigate to the desired folder

   ```bash
   cd na_cloud_volumes_automation/
   ```

3. Save the GCP authentication key JSON file in the directory.

4. Update the variable values in `vars/gcp_cvo_ha_deployment.tfvars`.

---

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcp_cvo_project_id</td>
<td>String</td>
<td>(Required) The ID of the GCP project.</td>
</tr>
<tr>
<td>gcp_cvo_zone</td>
<td>String</td>
<td>(Required) The zone of the region where the working environment will be created.</td>
</tr>
<tr>
<td>gcp_cvo_gcp_service_account</td>
<td>String</td>
<td>(Required) The gcp_service_account email in order to enable tiering of cold data to Google Cloud Storage.</td>
</tr>
<tr>
<td>gcp_cvo_svm_password</td>
<td>String</td>
<td>(Required) The admin password for Cloud Volumes ONTAP.</td>
</tr>
<tr>
<td>gcp_cvo_workspace_id</td>
<td>String</td>
<td>(Optional) The ID of the Cloud Manager workspace where you want to deploy Cloud Volumes ONTAP. If not provided, Cloud Manager uses the first workspace. You can find the ID from the Workspace tab on <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
<tr>
<td>gcp_cvo_capacity_package_name</td>
<td>String</td>
<td>(Optional) The capacity package name: ['Essential', 'Professional', 'Freemium']. Default is 'Essential'.</td>
</tr>
</tbody>
</table>
You can choose to deploy the connector by setting the variable "gcp_connector_deploy_bool" value to true/false.

e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

   `terraform init`

f. Verify the terraform files using terraform validate command.

   `terraform validate`

g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

   `terraform plan -target="module.gcp_ha" -var -file="vars/gcp_cvo_ha_deployment.tfvars"

h. Run the deployment

   `terraform apply -target="module.gcp_ha" -var -file="vars/gcp_cvo_ha_deployment.tfvars"

To delete the deployment

   `terraform destroy`

Recipes:
Connector

Terraform variables for NetApp GCP connector instance for CVO deployment.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcp_connector_deploy_bool</td>
<td>Bool</td>
<td>(Required) Check for Connector deployment.</td>
</tr>
<tr>
<td>gcp_connector_name</td>
<td>String</td>
<td>(Required) The name of the Cloud Manager Connector.</td>
</tr>
<tr>
<td>gcp_connector_project_id</td>
<td>String</td>
<td>(Required) The GCP project_id where the connector will be created.</td>
</tr>
<tr>
<td>gcp_connector_zone</td>
<td>String</td>
<td>(Required) The GCP zone where the Connector will be created.</td>
</tr>
</tbody>
</table>
### Name | Type | Description
--- | --- | ---
gcp_connector_company | String | (Required) The name of the company of the user.
gcp_connector_service_account_email | String | (Required) The email of the service_account for the connector instance. This service account is used to allow the Connector to create Cloud Volume ONTAP.
gcp_connector_service_account_path | String | (Required) The local path of the service_account JSON file for GCP authorization purposes. This service account is used to create the Connector in GCP.
gcp_connector_account_id | String | (Optional) The NetApp account ID that the Connector will be associated with. If not provided, Cloud Manager uses the first account. If no account exists, Cloud Manager creates a new account. You can find the account ID in the account tab of Cloud Manager at https://cloudmanager.netapp.com.

### HA Pair

Terraform variables for NetApp CVO instances in HA Pair on GCP.

### Name | Type | Description
--- | --- | ---
gcp_cvo_is_ha | Bool | (Optional) Indicate whether the working environment is an HA pair or not [true, false]. The default is false.
gcp_cvo_name | String | (Required) The name of the Cloud Volumes ONTAP working environment.
gcp_cvo_project_id | String | (Required) The ID of the GCP project.
gcp_cvo_zone | String | (Required) The zone of the region where the working environment will be created.
gcp_cvo_node1_zone | String | (Optional) Zone for node 1.
gcp_cvo_node2_zone | String | (Optional) Zone for node 2.
gcp_cvo_mediat_zone | String | (Optional) Zone for mediator.
gcp_cvo_vpc_id | String | (Optional) The name of the VPC.
gcp_cvo_subnet_id | String | (Optional) The name of the subnet for Cloud Volumes ONTAP. The default is: 'default'.
gcp_cvo_vpc0_node_and_data_connectivity | String | (Optional) VPC path for nic1, required for node and data connectivity. If using shared VPC, network_project_id must be provided.
gcp_cvo_vpc1_cluster_connectivity | String | (Optional) VPC path for nic2, required for cluster connectivity.
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcp_cvo_vpc2_ha_connectivity</td>
<td>String</td>
<td>(Optional) VPC path for nic3, required for HA connectivity.</td>
</tr>
<tr>
<td>gcp_cvo_vpc3_data_replication</td>
<td>String</td>
<td>(Optional) VPC path for nic4, required for data replication.</td>
</tr>
<tr>
<td>gcp_cvo_subnet0_node_and_data_connectivity</td>
<td>String</td>
<td>(Optional) Subnet path for nic1, required for node and data connectivity. If using shared VPC, network_project_id must be provided.</td>
</tr>
<tr>
<td>gcp_cvo_subnet1_cluster_connectivity</td>
<td>String</td>
<td>(Optional) Subnet path for nic2, required for cluster connectivity.</td>
</tr>
<tr>
<td>gcp_cvo_subnet2_ha_connectivity</td>
<td>String</td>
<td>(Optional) Subnet path for nic3, required for HA connectivity.</td>
</tr>
<tr>
<td>gcp_cvo_subnet3_data_replication</td>
<td>String</td>
<td>(Optional) Subnet path for nic4, required for data replication.</td>
</tr>
<tr>
<td>gcp_cvo_gcp_service_account</td>
<td>String</td>
<td>(Required) The gcp_service_account email in order to enable tiering of cold data to Google Cloud Storage.</td>
</tr>
<tr>
<td>gcp_cvo_svm_password</td>
<td>String</td>
<td>(Required) The admin password for Cloud Volumes ONTAP.</td>
</tr>
<tr>
<td>gcp_cvo_workspace_id</td>
<td>String</td>
<td>(Optional) The ID of the Cloud Manager workspace where you want to deploy Cloud Volumes ONTAP. If not provided, Cloud Manager uses the first workspace. You can find the ID from the Workspace tab on <a href="https://cloudmanager.netapp.com">https://cloudmanager.netapp.com</a>.</td>
</tr>
<tr>
<td>gcp_cvo_capacity_package_name</td>
<td>String</td>
<td>(Optional) The capacity package name: ['Essential', 'Professional', 'Freemium']. Default is 'Essential'.</td>
</tr>
<tr>
<td>gcp_cvo_gcp_volume_size</td>
<td>String</td>
<td>(Optional) The GCP volume size for the first data aggregate. For GB, the unit can be: [100 or 500]. For TB, the unit can be: [1,2,4,8]. The default is '1' .</td>
</tr>
<tr>
<td>gcp_cvo_gcp_volume_size_unit</td>
<td>String</td>
<td>(Optional) ['GB' or 'TB']. The default is 'TB'.</td>
</tr>
</tbody>
</table>

**CVS Volume**

Terraform configuration files for deployment of NetApp CVS Volume on GCP

This section contains various Terraform configuration files to deploy/configure NetApp CVS (Cloud Volumes Services) Volume on GCP (Google Cloud Platform).

Procedure
In order to run the template:

a. Clone the repository.

```
git clone https://github.com/NetApp/na_cloud_volumes_automation.git
```

b. Navigate to the desired folder

```
cd na_cloud_volumes_automation/
```

c. Save the GCP authentication key JSON file in the directory.

d. Update the variable values in `vars/gcp_cvs_volume.tfvars`.

e. Initialize the Terraform repository to install all the pre-requisites and prepare for deployment.

```
terraform init
```

f. Verify the terraform files using `terraform validate` command.

```
terraform validate
```

g. Make a dry run of the configuration to get a preview of all the changes expected by the deployment.

```
terraform plan -target="module.gcp_cvs_volume" -var -file="vars/gcp_cvs_volume.tfvars"
```

h. Run the deployment

```
terraform apply -target="module.gcp_cvs_volume" -var -file="vars/gcp_cvs_volume.tfvars"
```

To delete the deployment

```
terraform destroy
```

Recipes:
CVS Volume
Terraform variables for NetApp GCP CVS Volume.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcp_cvs_name</td>
<td>String</td>
<td>(Required) The name of the NetApp CVS volume.</td>
</tr>
<tr>
<td>gcp_cvs_project_id</td>
<td>String</td>
<td>(Required) The GCP project_id where the CVS Volume will be created.</td>
</tr>
<tr>
<td>gcp_cvs_gcp_service_account_path</td>
<td>String</td>
<td>(Required) The local path of the service_account JSON file for GCP authorization purposes. This service account is used to create the CVS Volume in GCP.</td>
</tr>
<tr>
<td>gcp_cvs_region</td>
<td>String</td>
<td>(Required) The GCP zone where the CVS Volume will be created.</td>
</tr>
<tr>
<td>gcp_cvs_network</td>
<td>String</td>
<td>(Required) The network VPC of the volume.</td>
</tr>
<tr>
<td>gcp_cvs_size</td>
<td>Integer</td>
<td>(Required) The size of volume is between 1024 to 102400 inclusive (in GiB).</td>
</tr>
<tr>
<td>gcp_cvs_volume_path</td>
<td>String</td>
<td>(Optional) The name of the volume path for volume.</td>
</tr>
<tr>
<td>gcp_cvs_protocol_types</td>
<td>String</td>
<td>(Required) The protocol_type of the volume. For NFS use 'NFSv3' or 'NFSv4' and for SMB use 'CIFS' or 'SMB'.</td>
</tr>
</tbody>
</table>

FSx for ONTAP Monitoring and Auto-Resizing using AWS Lambda Function

Overview: Monitoring and Auto-Resizing FSx for ONTAP via AWS Lambda function

FSx for ONTAP is a first-party enterprise-grade cloud storage service available on AWS that provides highly reliable, scalable, high-performing and feature-rich file storage built on the popular NetApp ONTAP file system.

FSx for ONTAP provides a seamless deployment and management experience. No storage expertise is required to get started. To simplify monitoring, an AWS lambda function (to automate resizing of total storage capacity, volume size or LUN size based on threshold) can be used. This document provides a step-by-step guide to creating an automated setup that monitors FSx for ONTAP at regular intervals, notifies and resizes when a user-specified threshold is crossed and notifies the administrator of the resizing activity.
Features

The solution provides the following features:

- Ability to monitor:
  - Usage of overall Storage Capacity of FSx for ONTAP
  - Usage of each volume (thin provisioned / thick provisioned)
  - Usage of each LUN (thin provisioned / thick provisioned)
- Ability to resize any of the above when a user-defined threshold is breached
- Alerting mechanism to receive usage warnings and resizing notifications via email
- Ability to delete snapshots older than the user-defined threshold
- Ability to get a list of FlexClone volumes and snapshots associated
- Ability to monitor the checks at a regular interval
- Ability to use the solution with or without internet access
- Ability to deploy manually or using AWS CloudFormation Template
- Ability to monitor multiple FSx for ONTAP filesystems in a single VPC

Pre-requisites

Before you begin, ensure that the following prerequisites are met:

- FSx for ONTAP is deployed
- Private subnet with connectivity to FSx for ONTAP
- "fsxadmin" password has been set for FSx for ONTAP
High-Level Architecture

- AWS Lambda Function makes API calls to FSx for ONTAP for retrieving and updating the size of Storage Capacity, Volumes and LUNs.
- "fsxadmin" password stored as secure string in AWS SSM Parameter Store for an added layer of security.
- AWS SES (Simple Email Service) is used to notify end-users when a resizing event occurs.
- If deploying the solution in a VPC without internet access, VPC Endpoints for AWS SSM, FSx and SES are set up to allow Lambda to reach these services via AWS internal network.
Solution Deployment

Automated Deployment

Covers single FSx for ONTAP filesystem.

Follow the series of steps to complete the automated deployment of this solution:

Step 1: Clone the GitHub repository

Clone the GitHub repository on your local system:

```
git clone https://github.com/NetApp/fsxn-monitoring-auto-resizing.git
```
Step 2: Setup an AWS S3 bucket

1. Navigate to AWS Console > S3 and click on **Create bucket**. Create the bucket with the default settings.
2. Once inside the bucket, click on **Upload > Add files** and select **Utilities.zip** from the cloned GitHub repository on your system.

![Image of AWS S3 console showing the upload process](image-url)
Step 3: AWS SES SMTP Setup (required if no internet access available)

Follow this step if you want to deploy the solution without internet access (Note: There will be added costs associated with VPC endpoints being set up.)

1. Navigate to AWS Console > AWS Simple Email Service (SES) > SMTP Settings and click on Create SMTP credentials

2. Enter an IAM User Name or leave it at the default value and click on Create User. Save the SMTP user name and SMTP password for further use.

Skip this step if SES SMTP setup is already in place.
Step 4: AWS CloudFormation Deployment

1. Navigate to AWS Console > **CloudFormation** > Create stack > With New Resources (Standard).

Prepare template: Template is ready
Specify template: Upload a template file
Choose file: Browse to the cloned GitHub repo and select `fsxn-monitoring-solution.yaml`

Click on Next

2. Enter the stack details. Click on Next and check the checkbox for "I acknowledge that AWS CloudFormation might create IAM resources" and click on Submit.

If "Does VPC have internet access?" is set to False, "SMTP Username for AWS SES" and "SMTP Password for AWS SES" are required. Otherwise, they can be left empty.
3. Once the CloudFormation deployment starts, the email ID mentioned in the "sender email ID" will get an email asking them to authorize the use of the email address with AWS SES. Click on the link to verify the email address.

4. Once the CloudFormation stack deployment is completed, if there are any warnings/notifications, an email will be sent to the recipient email ID with the notification details.
Manual Deployment

Supports monitoring multiple FSx for ONTAP filesystems in a single VPC.

Follow the series of steps to complete the manual deployment of this solution:

Step 1: Clone the GitHub repository

Clone the GitHub repository on your local system:

```
git clone https://github.com/NetApp/fsxn-monitoring-auto-resizing.git
```
Step 2: AWS SES SMTP Setup (required if no internet access available)

Follow this step if you want to deploy the solution without internet access (Note: There will be added costs associated with VPC endpoints being set up.)

1. Navigate to AWS Console > **AWS Simple Email Service (SES)** > SMTP Settings and click on **Create SMTP credentials**

2. Enter an IAM User Name or leave it at the default value and click on Create. Save the username and password for further use.
Step 3: Create SSM parameter for fsxadmin password

Navigate to AWS Console > **Parameter Store** and click on **Create Parameter**.

Name: <Any name/path for storing fsxadmin password>
Tier: Standard
Type: SecureString
KMS key source: My current account
KMS Key ID: <Use the default one selected>
Value: <Enter the password for "fsxadmin" user configured on FSx for ONTAP>

Click on **Create parameter**.
Repeat the above steps for all FSx for ONTAP filesystems to be monitored.

Perform the same steps for storing smtp username and smtp password if deploying the solution without internet access. Otherwise, skip adding these 2 parameters.
Step 4: Setup Email Service

Navigate to AWS Console > Simple Email Service (SES) and click on Create Identity.

Identity type: Email address
Email address: <Enter an email address to be used for sending resizing notifications>

Click on Create identity

The email ID mentioned in the "sender email ID" will get an email asking the owner to authorize the use of the email address with AWS SES. Click on the link to verify the email address.
Step 5: Setup VPC Endpoints (required if no internet access is available)

Required only if deployed without internet access. There will be additional costs associated with VPC endpoints.

1. Navigate to AWS Console > **VPC > Endpoints** and click on **Create Endpoint** and enter the following details:

   - **Name**: <Any name for the vpc endpoint>
   - **Service category**: AWS Services
   - **Services**: com.amazonaws.<region>.fsx
   - **vpc**: <select the vpc where lambda will be deployed>
   - **subnets**: <select the subnets where lambda will be deployed>
   - **Security groups**: <select the security group>
   - **Policy**: <Either choose Full access or set your own custom policy>

   Click on Create endpoint.
Create endpoint

There are three types of VPC endpoints – Interface endpoints, Gateway Load Balancer endpoints, and Gateway endpoints. Interface endpoints and Gateway Load Balancer endpoints are powered by AWS PrivateLink, and use an Elastic Network Interface (ENI) as an entry point for traffic destined to the service. Interface endpoints are typically accessed using the public or private DNS name associated with the service, while Gateway endpoints and Gateway Load Balancer endpoints serve as a target for a route in your route table for traffic destined for the service.

Endpoint settings

**Name tag - optional**
Creates a tag with a key of `Name` and a value that you specify.

```
fsxEndpoint
```

**Service category**
Select the service category

- **AWS services**
  Services provided by Amazon

- **PrivateLink Ready partner services**
  Services with an AWS Service Ready designation

- **AWS Marketplace services**
  Services that you’ve purchased through AWS Marketplace

- **Other endpoint services**
  Find services shared with you by service name

**Services (1/1)**

- **Service Name**
  `com.amazonaws.us-west-1.fsx`

- **Owner**
  `amazon`

- **Type**
  `Interface`
2. Follow the same process for creating SES and SSM VPC endpoints. All parameters remain the same as above except Services which will correspond to `com.amazonaws.<region>.smtp` and `com.amazonaws.<region>.ssm` respectively.
Step 6: Create and setup the AWS Lambda Function

1. Navigate to AWS Console > AWS Lambda > Functions and click on Create function in the same region as FSx for ONTAP

2. Use the default Author from scratch and update the following fields:

   - Function name: <Any name of your choice>
   - Runtime: Python 3.9
   - Architecture: x86_64
   - Permissions: Select "Create a new role with basic Lambda permissions"
   - Advanced Settings:
     - Enable VPC: Checked
     - VPC: <Choose either the same VPC as FSx for ONTAP or a VPC that can access both FSx for ONTAP and the internet via a private subnet>
     - Subnets: <Choose 2 private subnets that have NAT gateway attached pointing to public subnets with internet gateway and subnets that have internet access>
     - Security Group: <Choose a Security Group>

Click on Create function.
3. Navigate to the newly created Lambda function > Scroll down to the **Layers** section and click on **Add a layer**.

4. Click on **create a new layer** under **Layer source**

5. Create a Layer and upload **Utilities.zip** file. Select **Python 3.9** as the compatible runtime and click on **Create**.
6. Navigate back to AWS Lambda function > Add Layer > Custom Layers and add the utilities layer.
7. Navigate to the **Configuration** tab of the Lambda function and click on **Edit** under **General Configuration**. Change the Timeout to **5 mins** and click **Save**.

8. Navigate to **Permissions** tab of the Lambda function and click on the role assigned. In the permissions tab of the role, click on **Add permissions > Create Inline policy**.
   a. Click on the JSON tab and paste the contents of the file policy.json from the GitHub repo.
   b. Replace every occurrence of `${AWS::AccountId}` with your account ID and click on **Review Policy**
   c. Provide a Name for the policy and click on **Create policy**

9. Copy the contents of **fsxn_monitoring_resizing_lambda.py** from the git repo to **lambda_function.py** in the AWS Lambda function Code Source section.

10. Create a new file in the same level as lambda_function.py and name it **vars.py** and copy the contents of vars.py from the git repo to the lambda function vars.py file. Update the variable values in vars.py. Reference variable definitions below and click on **Deploy**:
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fsxList</td>
<td>List</td>
<td>(Required) List of all FSx for ONTAP filesystems to be monitored. Include all the file systems in the list for monitoring and auto-resizing.</td>
</tr>
<tr>
<td>fsxMgmtIp</td>
<td>String</td>
<td>(Required) Enter the &quot;Management endpoint - IP address&quot; from the FSx for ONTAP console on AWS.</td>
</tr>
<tr>
<td>fsxId</td>
<td>String</td>
<td>(Required) Enter the &quot;File system ID&quot; from the FSx for ONTAP console on AWS.</td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td>(Required) Enter the FSx for ONTAP &quot;ONTAP administrator username&quot; from FSx for ONTAP console on AWS.</td>
</tr>
<tr>
<td>resize_threshold</td>
<td>Integer</td>
<td>(Required) Enter the threshold percentage from 0-100. This threshold will be used to measure Storage Capacity, Volume and LUN usage and when the % use of any increases above this threshold, resize activity will occur.</td>
</tr>
<tr>
<td>fsx_password_ssm_parameter</td>
<td>String</td>
<td>(Required) Enter the path name used in AWS Parameter Store for storing &quot;fsxadmin&quot; password.</td>
</tr>
<tr>
<td>warn_notification</td>
<td>Bool</td>
<td>(Required) Set this variable to True to receive a notification when Storage Capacity/Volume/LUN usage exceeds 75% but is less than the threshold.</td>
</tr>
<tr>
<td>enable_snapshot_deletion</td>
<td>Bool</td>
<td>(Required) Set this variable to True to enable volume level snapshot deletion for snapshots older than the value specified in &quot;snapshot_age_threshold_in_days&quot;.</td>
</tr>
<tr>
<td>snapshot_age_threshold_in_days</td>
<td>Integer</td>
<td>(Required) Enter the number of days of volume level snapshots you want to retain. Any snapshots older than the value provided will be deleted and the same will be notified via email.</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>internet_access</td>
<td>Bool</td>
<td>(Required) Set this variable to True if internet access is available from the subnet where this lambda is deployed. Otherwise set it to False.</td>
</tr>
<tr>
<td>smtp_region</td>
<td>String</td>
<td>(Optional) If &quot;internet_access&quot; variable is set to False, enter the region in which lambda is deployed. E.g. us-east-1 (in this format)</td>
</tr>
<tr>
<td>smtp_password_ssm_parameter</td>
<td>String</td>
<td>(Optional) If &quot;internet_access&quot; variable is set to False, enter the path name used in AWS Parameter Store for storing the SMTP password.</td>
</tr>
<tr>
<td>smtp_username_ssm_parameter</td>
<td>String</td>
<td>(Optional) If &quot;internet_access&quot; variable is set to False, enter the path name used in AWS Parameter Store for storing the SMTP username.</td>
</tr>
<tr>
<td>sender_email</td>
<td>String</td>
<td>(Required) Enter the email ID registered on SES that will be used by the lambda function to send notification alerts related to monitoring and resizing.</td>
</tr>
<tr>
<td>recipient_email</td>
<td>String</td>
<td>(Required) Enter the email ID on which you want to receive the alert notifications.</td>
</tr>
</tbody>
</table>

11. Click on Test, create a test event with an empty JSON object and run the test by clicking Invoke to check if the script is running properly.

12. Once tested successfully, navigate to Configuration > Triggers > Add Trigger.
Select a Source: EventBridge
Rule: Create a new rule
Rule name: <Enter any name>
Rule type: Schedule expression
Schedule expression: <Use "rate(1 day)" if you want the function to run daily or add your own cron expression>

Click on Add.

Conclusion

With the provided solution, it is easy to set up a monitoring solution that regularly monitors FSx for ONTAP Storage, resizes it based on a user-specified threshold and provides an alerting mechanism. This makes the process of using and monitoring FSx for ONTAP seamless freeing up administrators to focus on business-critical activities while storage grows automatically when required.
Legal notices

Legal notices provide access to copyright statements, trademarks, patents, and more.

Copyright

https://www.netapp.com/company/legal/copyright/

Trademarks

NETAPP, the NETAPP logo, and the marks listed on the NetApp Trademarks page are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

https://www.netapp.com/company/legal/trademarks/

Patents

A current list of NetApp owned patents can be found at:


Privacy policy


Open source

Notice files provide information about third-party copyright and licenses used in NetApp software.