Hybrid Cloud with Provider-Managed Components
NetApp Solutions
NetApp
May 17, 2024
# Table of Contents

NetApp Hybrid Multicloud solutions for Red Hat OpenShift Container workloads ........................................... 1
   Overview .................................................................................................................................................. 1
Deploy and configure the Managed Red Hat OpenShift Container platform on AWS ................................. 4
Data protection ....................................................................................................................................... 7
Data migration ......................................................................................................................................... 23
NetApp Hybrid Multicloud solutions for Red Hat OpenShift Container workloads

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.

Overview

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.
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**
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:

![Storage Classes screenshot](image)

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.
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)

```bash
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:
oc login https://api.my-rosa-cluster.abcd.p1.openshiftapps.com:6443 \
--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 (Project:HelmChartRepository)
  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 vservname 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, 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
type: Secret
metadata:
  name: backend-tbc-ontap-nas-secret
type: Opaque
stringData:
  username: fsxadmin
  password: <password>
---
apiVersion: trident.netapp.io/v1
type: 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.

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.

```
[ec2-user@ip-10-49-11-132 astra]$ oc get project postgresql -o yaml | grep uid-range
openshift.io/sa.scc.uid-range: 1001010000/10000
[ec2-user@ip-10-49-11-132 astra]$ oc edit -n postgresql statefulset.apps/postgresql
```

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

```bash
[ec2-user@ip-10-49-11-132 astra]$ oc get pods -n postgresql
NAME       READY STATUS    AGE
postgresql-0 1/1 Running   0 2m46s
[ec2-user@ip-10-49-11-132 astra]$ kubectl get pvc -n postgresql
NAME           CAPACITY ACCESS MODES STORAGECLASS AGE
data-postgresql 8Gi RWO ontap-nas 4m2s
```

```bash
[ec2-user@ip-10-49-11-132 astra]$ export POSTGRES_PASSWORD=$(kubectl get secret --namespace postgresql postgresql -o jsonpath="{{.data.postgres-password}}") | base64 -d
[ec2-user@ip-10-49-11-132 astra]$ kubectl run postgresql-client --rm -tty -i --image docker.io/bitnami/postgresql:10.2.0-debian-10-r - -env "POSTGRES_PASSWORD=POSTGRES_PASSWORD"
> --command psql --host postgresql -U postgres -d postgres -p 5432
Warning: would violate PodSecurity "restricted:v1.24": allowPrivilegedEscalation (false), restricted capabilities (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# 
You are now connected to database "erp" as user "postgres".
```

```bash
erpc# CREATE TABLE PERSONS(id INT PRIMARY KEY NOT NULL, FIRSTNAME TEXT NOT NULL, LASTNAME TEXT NOT NULL);
CREATE TABLE
erpc# INSERT INTO PERSONS VALUES(1,'John','Doe');
INSERT 1 (1 row)
```

```bash
erpc# SELECT * FROM persons;
id | Firstname | Lastname
01  | John      | Doe
(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
added to ACS.

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.

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