# Table of Contents

NetApp Enterprise Database Solutions .................................................. 1
  Oracle Database .................................................................................. 1
Microsoft SQL Server ............................................................................. 493
Open Source Databases .......................................................................... 589
SnapCenter for Databases ..................................................................... 599
DB Automation Toolkits ......................................................................... 830
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].
Architecture

Oracle VLDB Incremental Merge via RMAN on ANF

Hardware and software components

<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>
<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.
Create a volume

Configure access to your volume.

Access
Protocol type
- NFS
- SMB
- Dual-protocol

Configuration
File path
ora-01-u02-copy

Versions
NFSv3

Kerberos
- Enabled
- Disabled

LDAP
- Enabled
- Disabled

Unix Permissions
0770

Azure VMware Solution DataStore

Export policy
Configure the volume's export policy. This can be edited later. Learn more ▼

Index | Allowed clients | Access | Root Access | Chown Mode
--- | --- | --- | --- | ---
1 | 172.30.137.128/25 | Read & Write | On | Restricted

Move up / Move down / Move to top / Move to bottom / Delete

Review + create

5. Add a volume tag if desired.
Create a volume

Tags

Tags are name/value pairs that enable you to categorize resources and view consolidated billing by applying the same tag to multiple resources and resource groups. Learn more about tags of.

Note that if you create tags and then change resource settings on other tabs, your tags will be automatically updated.

Name

Value

database
oracle

Review + create

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
```
8. Change mount point ownership to oracle:oinstall, change to your oracle user name and primary group as necessary.

```bash
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.

```
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.

```
sqlplus / as sysdba

From sqlplus prompt, execute following command.

```
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.

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.

connected 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.

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__04h19dgr_.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__04h19dgr_.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 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 Set</td>
<td>3</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>3</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 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>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>
Backup Set                  5      18-MAR-24
  Backup Piece               5      18-MAR-24
Backup Set                  6      18-MAR-24
  Backup Piece               6      18-MAR-24
Backup Set                  7      18-MAR-24
  Backup Piece               7      18-MAR-24
Backup Set                  8      18-MAR-24
  Backup Piece               8      18-MAR-24
Backup Set                  9      18-MAR-24
  Backup Piece               9      18-MAR-24
Backup Set                 10      18-MAR-24
  Backup Piece              10      18-MAR-24
Backup Set                 11      18-MAR-24
  Backup Piece              11      18-MAR-24
Backup Set                 12      18-MAR-24
  Backup Piece              12      18-MAR-24
deleted backup piece
backup piece
handle=/u03/orareco/NTAP1/autobackup/2024_03_18/o1_mf_s_1163963796__09k8g1m4_.bkp RECID=4 STAMP=1163963804
deleted backup piece
backup piece
handle=/u03/orareco/NTAP1/autobackup/2024_03_18/o1_mf_s_1163962888__08p6y71x_.bkp RECID=3 STAMP=1163962897
deleted backup piece
backup piece
handle=/u03/orareco/NTAP1/autobackup/2024_03_18/o1 mf s_1163961675__
0711m2lg_.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__
0bd3tgq3_.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__
0dbx344_.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

deleated 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

deleated 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=9
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=10
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=11
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_0s2m6n11 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-
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
channel c4: starting datafile copy
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
channel c1: starting datafile copy
input datafile file number=00009
name=/u02/oradata/NTAP1/NTAP1_pdb1/system01.dbf
output file name=/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
UNDOTBS1_FNO-11_0u2m6nqs tag=ORACOPYBKUPONANF_LEVEL_0 RECID=21
STAMP=1164140471
channel c2: datafile copy complete, elapsed time: 00:01:33
channel c2: starting datafile copy
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
channel c3: starting datafile copy
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
channel c4: starting datafile copy
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
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';
```

<table>
<thead>
<tr>
<th>Key</th>
<th>File S 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>20-MAR-24</td>
<td>4161498</td>
<td>NO</td>
</tr>
</tbody>
</table>
|     | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
|     | SYSTEM_FNO-1_0r2m6nhk|
|     | Tag: ORACOPYBKUPONANF_LEVEL_0 |
| 16  | 3                    | 20-MAR-24| 4161568     | NO     |
|     | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
|     | SYSTEM_FNO-16_1c2m6o0k|
|     | Tag: ORACOPYBKUPONANF_LEVEL_0 |
```
SYSAUX_FNO-3_0s2m6nl1
Tag: ORACOPYBKUPONANF_LEVEL_0

15  4  A 20-MAR-24    4161589  20-MAR-24    NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
UNDOTBS1_FNO-4_0t2m6nml
Tag: ORACOPYBKUPONANF_LEVEL_0

27  5  A 20-MAR-24    2379694  18-MAR-24    NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SYSTEM_FNO-5_162m6nuc
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 2, PDB Name: PDB$SEED

23  6  A 20-MAR-24    2379694  18-MAR-24    NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SYSAUX_FNO-6_122m6nti
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 2, PDB Name: PDB$SEED

29  7  A 20-MAR-24    4161872  20-MAR-24    NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
USERS_FNO-7_1a2m6o01
Tag: ORACOPYBKUPONANF_LEVEL_0

28  8  A 20-MAR-24    2379694  18-MAR-24    NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
UNDOTBS1_FNO-8_172m6nvr
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 2, PDB Name: PDB$SEED

26  9  A 20-MAR-24    4161835  20-MAR-24    NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SYSTEM_FNO-9_132m6ntm
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 3, PDB Name: NTAP1_PDB1

19 10  A 20-MAR-24    4161784  20-MAR-24    NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
SYSAUX_FNO-10_0v2m6nqs
Tag: ORACOPYBKUPONANF_LEVEL_0
Container ID: 3, PDB Name: NTAP1_PDB1

21 11  A 20-MAR-24    4161780  20-MAR-24    NO
Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-
UNDOTBS1_FNO-11_0u2m6nqs
<table>
<thead>
<tr>
<th>No.</th>
<th>ID</th>
<th>Type</th>
<th>Date</th>
<th>Time</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>12</td>
<td>A</td>
<td>20-MAR-24</td>
<td>4161880</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-</td>
</tr>
<tr>
<td>24</td>
<td>13</td>
<td>A</td>
<td>20-MAR-24</td>
<td>4161838</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>A</td>
<td>20-MAR-24</td>
<td>4161785</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>A</td>
<td>20-MAR-24</td>
<td>4161863</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-</td>
</tr>
<tr>
<td>34</td>
<td>16</td>
<td>A</td>
<td>20-MAR-24</td>
<td>4161884</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-</td>
</tr>
<tr>
<td>25</td>
<td>17</td>
<td>A</td>
<td>20-MAR-24</td>
<td>4161841</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-</td>
</tr>
<tr>
<td>22</td>
<td>18</td>
<td>A</td>
<td>20-MAR-24</td>
<td>4161810</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-</td>
</tr>
<tr>
<td>31</td>
<td>19</td>
<td>A</td>
<td>20-MAR-24</td>
<td>4161869</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-</td>
</tr>
<tr>
<td>Name</td>
<td>Container ID</td>
<td>PDB Name</td>
<td>Tag</td>
<td>Date</td>
<td>Size</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------</td>
<td>-----------</td>
<td>------------------------------------------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-UNDOTBS1_FNO-19_192m6nvv</td>
<td>5</td>
<td>NTAP1_PDB3</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>20-MAR-24</td>
<td>4161887</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- USERS_FNO-20_1d2m6o0k</td>
<td>5</td>
<td>NTAP1_PDB3</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>20-MAR-24</td>
<td>4152514</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-21_0f2m6brl</td>
<td>3</td>
<td>NTAP1_PDB1</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>20-MAR-24</td>
<td>4152518</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-22_0g2m6brl</td>
<td>3</td>
<td>NTAP1_PDB1</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>20-MAR-24</td>
<td>4152522</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-23_0h2m6brl</td>
<td>3</td>
<td>NTAP1_PDB1</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>20-MAR-24</td>
<td>4152529</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-24_0i2m6brl</td>
<td>3</td>
<td>NTAP1_PDB1</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>20-MAR-24</td>
<td>4156120</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-25_0j2m6fol</td>
<td>3</td>
<td>NTAP1_PDB1</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>20-MAR-24</td>
<td>4156130</td>
</tr>
<tr>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-26_0k2m6fot</td>
<td>3</td>
<td>NTAP1_PDB1</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>20-MAR-24</td>
<td>4156130</td>
</tr>
</tbody>
</table>
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
==================================
File Size(MB) Tablespace RB segs Datafile Name

<table>
<thead>
<tr>
<th>No.</th>
<th>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-27_0l2m6frc</th>
<th>Tag: ORACOPYBKUPONANF_LEVEL_0</th>
<th>Container ID: 3, PDB Name: NTAP1_PDB1</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>27 A 20-MAR-24 4156159 20-MAR-24 NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>28 A 20-MAR-24 4156183 20-MAR-24 NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>29 A 20-MAR-24 4158795 20-MAR-24 NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>30 A 20-MAR-24 4158803 20-MAR-24 NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>31 A 20-MAR-24 4158871 20-MAR-24 NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>32 A 20-MAR-24 4158886 20-MAR-24 NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<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>1</td>
<td>TEMP</td>
<td>32767</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PDB$SEED:TEMP</td>
<td>32767</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>NTAP1_PDB1:TEMP</td>
<td>32767</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>NTAP1_PDB2:TEMP</td>
<td>32767</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>NTAP1_PDB3:TEMP</td>
<td>32767</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>NTAP1_PDB1:TEMP</td>
<td>31744</td>
<td></td>
</tr>
</tbody>
</table>

RMAN>

9. Validate database image copy from OS NFS mount point.
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;

ID
--------

DT
----------------------------------------
------

EVENT
----------------------------------------
------

1
21-MAR-24 03.15.03.000000 PM
test oracle incremental merge switch to copy

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.

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-USER_S_FNO-7_1a2m6o01"
datafile 8 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-UNDOTBS1_FNO-8_172m6nvr"
datafile 9 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSTEM_FNO-9_132m6ntm"
datafile 10 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSAUX_FNO-10_0v2m6nqs"
datafile 11 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-UNDOTBS1_FNO-11_0u2m6nqs"
datafile 12 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-USER_S_FNO-12_1b2m6o0e"
datafile 13 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSTEM_FNO-13_142m6ntp"
datafile 14 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSAUX_FNO-14_102m6nr3"
datafile 15 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-UNDOTBS1_FNO-15_182m6nvs"
datafile 16 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-USER_S_FNO-16_1c2m6o0k"
datafile 17 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSTEM_FNO-17_152m6nts"
datafile 18 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SYSAUX_FNO-18_112m6nrt"
datafile 19 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-UNDOTBS1_FNO-19_192m6nvv"
datafile 20 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-USER_S_FNO-20_1d2m6o0k"
datafile 21 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SOE_FNO-21_0f2m6brl"
datafile 22 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-SOE_FNO-22_0g2m6brl"
datafile 23 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_012m6brl"
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_012m6frc"
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
cchannel 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_0q1sd7cm
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: restored 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> 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> /
NAME
-----------------------------------------------
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-21_0f2m6brl
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-22_0g2m6brl
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-23_0h2m6brl
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-24_0i2m6brl
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-25_0j2m6fol
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-26_0k2m6fot
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-27_0l2m6frc
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-28_0m2m6fsu
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-29_0n2m6jlr
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-30_0o2m6jlr
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-31_0p2m6jrb
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-32_0q2m6jsi
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-10_0v2m6nqs
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-14_102m6nr3
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-18_112m6nrt
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-3_0s2m6n1l
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-6_122m6nti
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-13_142m6ntp
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-17_152m6nts
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-1_0r2m6nhk
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-5_162m6nuc
/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-9_132m6ntm
```
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;

<table>
<thead>
<tr>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
</tr>
<tr>
<td>EVENT</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

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.
SQL> insert into test values(2, sysdate, 'test recovery on a new Azure VM host with image copy on ANF');

1 row created.

SQL> commit;

Commit complete.

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

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.

```
sudo mkdir /nfsanf
```

As azureuser, mount the NFS volume that stored Oracle VLDB backup image copy.

```
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.

```
[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.
[oracle@ora-02 ~]$ rman target / nocatalog

Recovery Manager: Release 19.0.0.0.0 - Production on Fri Mar 22 16:02:55 2024
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

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.

```sql
RMAN> restore spfile to pfile '/tmp/archive/initNTAP1.ora' from autobackup;
```

```sql
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
channel ORA_DISK_1: AUTOBACKUP
/nfsanf/archlog/NTAP1/autobackup/2024_03_22/o1_mf_s_1164296325__9z77 zyxh_.bkp 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__9z77 zyxh_.bkp
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.

```sql
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>
</table>
| 82  | 1      | A 22-MAR-24     | 4598427 | 22-MAR-24| NO     | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- 
      |        |                 |         |          |        | SYSTEM_FNO-1_1t2m9nij | Tag: ORACOPYBKUPONANF_LEVEL_0 |
| 83  | 3      | A 22-MAR-24     | 4598423 | 22-MAR-24| NO     | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- 
      |        |                 |         |          |        | SYSAUX_FNO-3_1u2m9nog  | Tag: ORACOPYBKUPONANF_LEVEL_0 |
| 84  | 4      | A 22-MAR-24     | 4598431 | 22-MAR-24| NO     | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- 
      |        |                 |         |          |        | UNDOTBS1_FNO-4_1v2m9nu6 | Tag: ORACOPYBKUPONANF_LEVEL_0 |
| 58  | 5      | A 21-MAR-24     | 2379694 | 18-MAR-24| NO     | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- 
      |        |                 |         |          |        | SYSTEM_FNO-5_282m9oem  | Tag: ORACOPYBKUPONANF_LEVEL_0 |
      |        |                 |         |          |        | Container ID: 2, PDB Name: PDB$SEED | |
| 52  | 6      | A 21-MAR-24     | 2379694 | 18-MAR-24| NO     | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- 
      |        |                 |         |          |        | SYSAUX_FNO-6_242m9oan   | Tag: ORACOPYBKUPONANF_LEVEL_0 |
      |        |                 |         |          |        | Container ID: 2, PDB Name: PDB$SEED | |
| 90  | 7      | A 22-MAR-24     | 4598462 | 22-MAR-24| NO     | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- 
      |        |                 |         |          |        | USERS_FNO-7_2c2m9ofn    | Tag: ORACOPYBKUPONANF_LEVEL_0 |
| 59  | 8      | A 21-MAR-24     | 2379694 | 18-MAR-24| NO     | Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS- 
<pre><code>  |        |                 |         |          |        | UNDOTBS1_FNO-8_292m9oem  | Tag: ORACOPYBKUPONANF_LEVEL_0 |
</code></pre>
<table>
<thead>
<tr>
<th>Container ID</th>
<th>PDB Name</th>
<th>Name</th>
<th>Tag</th>
<th>Container ID</th>
<th>PDB Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-9_252m9oc5</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>3</td>
<td>NTAP1_PDB1</td>
</tr>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-9_252m9oc5</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>3</td>
<td>NTAP1_PDB1</td>
</tr>
<tr>
<td>3</td>
<td>NTAP1_PDB1</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSAUX_FNO-10_212m9o52</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>3</td>
<td>NTAP1_PDB1</td>
</tr>
<tr>
<td>3</td>
<td>NTAP1_PDB1</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-UNDOTBS1_FNO-11_202m9o22</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>3</td>
<td>NTAP1_PDB1</td>
</tr>
<tr>
<td>3</td>
<td>NTAP1_PDB1</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-USERS_FNO-12_2d2m9ofs</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>3</td>
<td>NTAP1_PDB1</td>
</tr>
<tr>
<td>4</td>
<td>NTAP1_PDB2</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-13_262m9oca</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>4</td>
<td>NTAP1_PDB2</td>
</tr>
<tr>
<td>4</td>
<td>NTAP1_PDB2</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SYSTEM_FNO-13_262m9oca</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>4</td>
<td>NTAP1_PDB2</td>
</tr>
<tr>
<td>4</td>
<td>NTAP1_PDB2</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-UNDOTBS1_FNO-15_2a2m9of6</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>4</td>
<td>NTAP1_PDB2</td>
</tr>
<tr>
<td>4</td>
<td>NTAP1_PDB2</td>
<td>/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-UNDOTBS1_FNO-15_2a2m9of6</td>
<td>ORACOPYBKUPONANF_LEVEL_0</td>
<td>4</td>
<td>NTAP1_PDB2</td>
</tr>
</tbody>
</table>
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: 4, PDB Name: NTAP1_PDB2

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
<table>
<thead>
<tr>
<th>80</th>
<th>24</th>
<th>A</th>
<th>22-MAR-24</th>
<th>4598313</th>
<th>22-MAR-24</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-24_1k2m9cap</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag: ORACOPYBKUPONANF_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID: 3, PDB Name: NTAP1_PDB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>79</th>
<th>25</th>
<th>A</th>
<th>22-MAR-24</th>
<th>4598318</th>
<th>22-MAR-24</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-25_1l2m9g3u</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag: ORACOPYBKUPONANF_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID: 3, PDB Name: NTAP1_PDB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>69</th>
<th>26</th>
<th>A</th>
<th>22-MAR-24</th>
<th>4598304</th>
<th>22-MAR-24</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-26_1m2m9g9j</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag: ORACOPYBKUPONANF_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID: 3, PDB Name: NTAP1_PDB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>70</th>
<th>27</th>
<th>A</th>
<th>22-MAR-24</th>
<th>4598308</th>
<th>22-MAR-24</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-27_1n2m9gcg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag: ORACOPYBKUPONANF_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID: 3, PDB Name: NTAP1_PDB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>75</th>
<th>28</th>
<th>A</th>
<th>22-MAR-24</th>
<th>4598313</th>
<th>22-MAR-24</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-28_1o2m9gd4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag: ORACOPYBKUPONANF_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID: 3, PDB Name: NTAP1_PDB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>77</th>
<th>29</th>
<th>A</th>
<th>22-MAR-24</th>
<th>4598318</th>
<th>22-MAR-24</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-29_1p2m9ju6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag: ORACOPYBKUPONANF_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID: 3, PDB Name: NTAP1_PDB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>67</th>
<th>30</th>
<th>A</th>
<th>22-MAR-24</th>
<th>4598304</th>
<th>22-MAR-24</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-30_1q2m9k7a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag: ORACOPYBKUPONANF_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID: 3, PDB Name: NTAP1_PDB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>76</th>
<th>31</th>
<th>A</th>
<th>22-MAR-24</th>
<th>4598308</th>
<th>22-MAR-24</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: /nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-31_1r2m9kfk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag: ORACOPYBKUPONANF_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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-UNDOHORT_FNO-7_2c2m9ofn"
datafile 8 switched to datafile copy "/nfsanf/oracopy/data_D-
NTAP1_I-2441823937_TS-UNDOHORT_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-UNDOHORT_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-UNDOHORT_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_272m9oe1"
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-UNDOHORT_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"
datafile 30 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-30_1q2m9k7a"
datafile 31 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-31_1r2m9kfk"
datafile 32 switched to datafile copy "/nfsanf/oracopy/data_D-NTAP1_I-2441823937_TS-SOE_FNO-32_1s2m9kgg"

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__9ybjdtp55_.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__9yomybhc_.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
media recovery complete, elapsed time: 00:01:17
Finished recover at 22-MAR-24

RMAN> exit

Recovery Manager complete.

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
    ---------------
    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;
31 rows selected.

SQL> select member from v$logfile;

MEMBER
-----------------------------------------------
---------
/nfsanf/oracopy/redo03.log
/nfsanf/oracopy/redo02.log
/nfsanf/oracopy/redo01.log

SQL> select name from v$controlfile;

NAME
-----------------------------------------------
---------
/nfsanf/oracopy/NTAP1.ctl

SQL> alter session set container=ntap1_pdb1;
Session altered.

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>
<tr>
<td>2</td>
<td>22-MAR-24 02.22.06.000000 PM</td>
<td>test recovery on a new Azure VM host with image copy on ANF</td>
</tr>
</tbody>
</table>

20. Drop invalid tempfiles and add new tempfiles to temp tablespaces.

SQL> select name from v$tempfile;
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
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=262144,wsize=262144,noin
tr 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>Description</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>Description</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"
   ```
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
[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 ###

# Database domain name
db_domain: solutions.netapp.com

# Set 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
There are a total of five playbooks in the automation toolkit. Each performs different task blocks and serves different purposes.

<table>
<thead>
<tr>
<th>Playbook Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-all_playbook.yml</td>
<td>Execute playbooks from 1-4 in one playbook run.</td>
</tr>
<tr>
<td>1-ansible_requirements.yml</td>
<td>Set up Ansible controller with required libs and collections.</td>
</tr>
<tr>
<td>2-linux_config.yml</td>
<td>Execute Linux kernel configuration on Oracle DB servers.</td>
</tr>
<tr>
<td>4-oracle_config.yml</td>
<td>Install and configure Oracle on DB servers and create a container database.</td>
</tr>
<tr>
<td>5-destroy.yml</td>
<td>Optional to undo the environment to dismantle all.</td>
</tr>
</tbody>
</table>

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 azureuser -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 azureuser -e @vars/vars.yml
   ```

   ```
   ansible-playbook -i hosts 2-linux_config.yml -u azureuser -e @vars/vars.yml
   ```

   ```
   ansible-playbook -i hosts 4-oracle_config.yml -u azureuser -e @vars/vars.yml
   ```

3. Execute 0-all_playbook.yml with a tag.

   ```
   ansible-playbook -i hosts 0-all_playbook.yml -u azureuser -e @vars/vars.yml -t ansible_requirements
   ```

   ```
   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
```
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

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...
```
Services Summary...
Service "104409ac02da6352e063bb891eacf34a.solutions.netapp.com" has 1 instance(s).
  Instance "NTAP1", status READY, has 1 handler(s) for this service...
Service "104412c14c2c63cae063bb891eacf64d.solutions.netapp.com" has 1 instance(s).
  Instance "NTAP1", status READY, has 1 handler(s) for this service...
Service "1044174670ad63ffe063bb891eac6b34.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;

<table>
<thead>
<tr>
<th>NAME</th>
<th>OPEN_MODE</th>
<th>LOG_MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTAP1</td>
<td>READ WRITE</td>
<td>ARCHIVELOG</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$SEED</td>
<td>READ ONLY</td>
<td>NO</td>
</tr>
<tr>
<td>3</td>
<td>NTAP1_PDB1</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
<tr>
<td>4</td>
<td>NTAP1_PDB2</td>
<td>READ WRITE</td>
<td>NO</td>
</tr>
</tbody>
</table>
SQL> select name from v$controlfile;

NAME
------------------------------------------
---
/u02/oradata/NTAP1/control01.ctl
/u03/orareco/NTAP1/control02.ctl

SQL> select member from v$logfile;

MEMBER
------------------------------------------
---
/u03/orareco/NTAP1/onlinelog/redo03.log
SQL> select svrname, dirname, nfsversion from v$dnfs_servers;

<table>
<thead>
<tr>
<th>SVRNAME</th>
<th>DIRNAME</th>
<th>NFSVERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.30.136.68</td>
<td>/ora-01-u02</td>
<td>NFSv3.0</td>
</tr>
<tr>
<td>172.30.136.68</td>
<td>/ora-01-u03</td>
<td>NFSv3.0</td>
</tr>
<tr>
<td>172.30.136.68</td>
<td>/ora-01-u01</td>
<td>NFSv3.0</td>
</tr>
</tbody>
</table>

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>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware</strong></td>
<td></td>
</tr>
<tr>
<td>Amazon FSx ONTAP storage</td>
<td>Current version offered by AWS</td>
</tr>
<tr>
<td>EC2 instance for compute</td>
<td>t2.xlarge/4vCPU/16G</td>
</tr>
<tr>
<td><strong>Software</strong></td>
<td></td>
</tr>
<tr>
<td>RedHat Linux</td>
<td>RHEL-8.6, 4.18.0-372.9.1.el8.x86_64 kernel</td>
</tr>
<tr>
<td>Windows Server</td>
<td>2022 Standard, 10.0.20348 Build 20348</td>
</tr>
</tbody>
</table>

See the section [Key Factors for Deployment Consideration] for more information.
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.

   ```
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.

   ```
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.

   ```
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:
Automated 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"

### Oracle DB env specific config variables

# 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.

```bash
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.

```bash
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 -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 ~]$ sudo su
[ec2-user@ip-172-30-15-40 ~]$ 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
```
+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/NTAP1/0C03ACEABA54D386E063280F1EACE573/DATAFILE/system.282.1155057091
+DATA/NTAP1/0C03ACEABA54D386E063280F1EACE573/DATAFILE/sysaux.283.1155057091
+DATA/NTAP1/0C03ACEABA54D386E063280F1EACE573/DATAFILE/undotbs1.281.1155057091
+DATA/NTAP1/0C03ACEABA54D386E063280F1EACE573/DATAFILE/users.285.1155057105

19 rows selected.

SQL> select name from v$controlfile;

NAME
2. Validate Oracle listener.

```
[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...

<table>
<thead>
<tr>
<th>Description</th>
<th>Address Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=ip-172-30-15-40.ec2.internal)(PORT=1521)))</td>
<td></td>
</tr>
<tr>
<td>(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))</td>
<td></td>
</tr>
<tr>
<td>(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=HT TP) (Session=RAW))</td>
<td></td>
</tr>
</tbody>
</table>

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 "0c03aafa7c6fd2e5e063280f1eacfb60.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).
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>------------------</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>Started, STABLE</td>
</tr>
<tr>
<td>Started,STABLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ora.ons</td>
<td>OFFLINE</td>
<td>OFFLINE</td>
<td>ip-172-30-15-40</td>
<td>STABLE</td>
</tr>
<tr>
<td>Cluster Resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>--------------</td>
<td>----------</td>
<td>-----------------------</td>
<td>----------</td>
</tr>
<tr>
<td>ora.cssd</td>
<td>1</td>
<td>ONLINE</td>
<td>ip-172-30-15-40</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.diskmon</td>
<td>1</td>
<td>OFFLINE</td>
<td></td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.driver.afd</td>
<td>1</td>
<td>ONLINE</td>
<td>ip-172-30-15-40</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.evmd</td>
<td>1</td>
<td>ONLINE</td>
<td>ip-172-30-15-40</td>
<td>STABLE</td>
</tr>
<tr>
<td>ora.ntap1.db</td>
<td>1</td>
<td>ONLINE</td>
<td>ip-172-30-15-40</td>
<td>STABLE</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  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.
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-8225-4784bced9843&sc_channel=ps&s_kwcid=AL!4422!3!467723097970le!!g!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKArIsAJsSrJdqwQrghn6i171jiWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:G:s&s_kwcid=AL!4422!3!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
Hardware and software components

### Hardware

<table>
<thead>
<tr>
<th>Component</th>
<th>Version/Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetApp ASA A400</td>
<td>Version 9.13.1P1</td>
<td>2 NS224 shelves, 48 NVMe AFF drives with total 69.3 TiB capacity</td>
</tr>
<tr>
<td>UCSB-B200-M4</td>
<td>Intel® Xeon® CPU E5-2690 v4 @ 2.60GHz</td>
<td>4-node VMware ESXi cluster</td>
</tr>
</tbody>
</table>

### Software

<table>
<thead>
<tr>
<th>Component</th>
<th>Version/Model</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>
<td>Deployed RedHat subscription for testing</td>
</tr>
<tr>
<td>Windows Server</td>
<td>2022 Standard, 10.0.20348 Build 20348</td>
<td>Hosting SnapCenter server</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 Server</td>
<td>Version 4.9P1</td>
<td>Workgroup deployment</td>
</tr>
<tr>
<td>VMware vSphere Hypervisor</td>
<td>version 6.5.0.20000</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

```
##### Oracle 19c deployment global user configurable variables ######
##### Consolidate all variables from ONTAP, linux and oracle #####
# Enter the supported ONTAP platform: on-prem, aws-fsx.
ontap_platform: on-prem

# Enter ONTAP cluster management user credentials
username: "xxxxxxxx"
password: "xxxxxxxx"

##### on-prem platform specific user defined variables #####

# 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.**

   ```bash
   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.**

   ```bash
   ansible-playbook -i hosts 1-ansible_requirements.yml -u admin -e @vars/vars.yml
   ```

   ```bash
   ansible-playbook -i hosts 2-linux_config.yml -u admin -e @vars/vars.yml
   ```

   ```bash
   ansible-playbook -i hosts 3-ontap_config.yml -u admin -e @vars/vars.yml
   ```

   ```bash
   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/sda1 1014M 258M 757M 26% /boot
tmpfs 1.6G 12K 1.6G 1% /run/user/1000
tmpfs 1.6G 4.0K 1.6G 1% /run/user/42/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
ora.DATA.dg ONLINE ONLINE ora_01 STABLE
ora.LISTENER.lsnr ONLINE INTERMEDIATE ora_01 Not All
ora.LOGS.dg ONLINE ONLINE ora_01 STABLE
ora.asm ONLINE ONLINE ora_01
ora.ons OFFLINE OFFLINE ora_01 STABLE
```
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 ~]$
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
# Oracle Database Deployment in VMware Cloud on AWS with Amazon FSx ONTAP

## 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 ONTAP HA cluster 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>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>
</tbody>
</table>
BlueXP backup and recovery for VMs | Release 1.0 | Deployed as an ova vSphere plugin VM
---|---|---
VMware vSphere | Version 8.0.1.00300 | VMware Tools, Version: 11365 - Linux, 12352 - Windows
Open JDK | Version java-1.8.0-openjdk.x86_64 | 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.

   ![Screenshot of VMware vSphere Client showing VMs](image)

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        589145 Oct  4 17:04 netapp_linux_unified_host_utilities-7-1.x86_64.rpm
   -rw-rw-r--. 1 admin  admin         31828 Oct  4 17:04 oracle-database-preinstall-19c-1.0-2.el8.x86_64.rpm
   -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        257136 Oct  4 17:04 policycoreutils-python-utils-2.9-9.el8.noarch.rpm
   [admin@ora_02 ~]$ 
   ```

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.

```bash
yum install /tmp/archive/compat-libcap1-1.10-7.el7.x86_64.rpm
```

5. From NetApp, download and install NetApp host utilities.

```bash
yum install /tmp/archive/netapp_linux_unified_host_utilities-7-1.x86_64.rpm
```


```bash
yum install /tmp/archive/policycoreutils-python-utils-2.9-9.el8.noarch.rpm
```

7. Install open JDK version 1.8.

```bash
yum install java-1.8.0-openjdk.x86_64
```

8. Install iSCSI initiator utils.

```bash
yum install iscsi-initiator-utils
```

9. Install `sg3_utils`.

```bash
yum install sg3_utils
```

10. Install `device-mapper-multipath`.

```bash
yum install device-mapper-multipath
```

11. Disable transparent hugepages in the current system.

```bash
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.
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.

    cat /etc/iscsi/initiatorname.iscsi

19. Add the asm groups for asm management user (oracle).

    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).

    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.

    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.

    chmod 640 /etc/sudoers
    vi /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.

   ssh fsxadmin@10.49.0.74

2. Execute the following command to create a volume for the Oracle binary.

   ```sh
   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.

   ```sh
   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.

   ```sh
   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.

   ```sh
   vol show ora*
   ```

   Output from the command:

   ```none
   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.
```
lun create -path /vol/ora_02_biny/ora_02_biny_01 -size 40G -ostype linux
```

7. Create data LUNs within the database data volume.
```
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.
```
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.
```
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>nim</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>
<tr>
<td>----------</td>
<td>------</td>
<td>--------</td>
<td>--------</td>
</tr>
</tbody>
</table>

126
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.

```
sudo iscsiadm iscsiadm --mode discovery --op update --type sendtargets --portal 10.49.0.12
```

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
Logging in to [iface: default, target: iqn.1992-08.com.netapp:sn.1f795e65c74911edeb785affbf0a2b26e:vs.3, portal: 10.49.0.12,3260]
```

4. View and validate a list of active iSCSI sessions.

```
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
controller(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_01
/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_01
/dev/sdf host33 iSCSI 20g cDOT
nim /vol/ora_02_data/ora_02_u02_01
/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 /devmapper 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 20:13 3600a09806c574235472455534e685146 -> ../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:
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_02_biny_01

12. Format the partitioned binary LUN with an XFS file system.

   sudo mkfs.xfs /dev/mapper/ora_02_biny_01p1

13. Mount the binary LUN to /u01.

   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 it's associated primary group.

   sudo chown oracle:oinstall /u01

15. Find the UUI of the binary LUN.

   sudo blkid /dev/mapper/ora_02_biny_01p1
16. Add a mount point to `/etc/fstab`.

```bash
sudo vi /etc/fstab
```

Add the following line.

```bash
UUID=d89fb1c9-4f89-4de4-b4d9-17754036d11d /u01 xfs defaults,nofail 0 2
```

17. As the root user, add the udev rule for Oracle devices.

```bash
vi /etc/udev/rules.d/99-oracle-asmdevices.rules
```

Include following entries:

```bash
ENV{DM_NAME}=="ora*", GROUP="oinstall", OWNER="oracle", MODE="660"
```

18. As the root user, reload the udev rules.

```bash
udevadm control --reload-rules
```

19. As the root user, trigger the udev rules.

```bash
udevadm trigger
```

20. As the root user, reload multipathd.

```bash
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`.

   ```bash
   sudo vi /etc/ssh/sshd_config
   ```

2. Restart the sshd service.

   ```bash
   sudo systemctl restart sshd
   ```

3. Reset the Oracle user password.

   ```bash
   sudo passwd oracle
   ```

4. Log in as the Oracle Restart software owner user (oracle). Create an Oracle directory as follows:

   ```bash
   mkdir -p /u01/app/oracle
   mkdir -p /u01/app/oraInventory
   ```

5. Change the directory permission setting.

   ```bash
   chmod -R 775 /u01/app
   ```

6. Create a grid home directory and change to it.

   ```bash
   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.

   ```bash
   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:

    ```bash
    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/
    ```

    ```bash
    export ORACLE_BASE=/tmp
    ```

    ```bash
    cd /u01/app/oracle/product/19.0.0/grid/bin
    ```
13. Initialize disk devices for use with the Oracle ASM filter driver.

```
./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`.

```
rpm -ivh /u01/app/oracle/product/19.0.0/grid/cv/rpm/cvuqdisk-1.0.10-1.rpm
```

15. Unset `$ORACLE_BASE`.

```
unset ORACLE_BASE
```

16. Log into the EC2 instance as the Oracle user and extract the patch in the `/tmp/archive` folder.

```
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.

```
./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
```
23. Validate ASM filter driver status.
24. Validate HA service status.

[oracle@ora_02 bin]$ ./crsctl check has
CRS-4638: Oracle High Availability Services is online
1. Log in as the Oracle user and unset $ORACLE_HOME and $ORACLE_SID if it is set.

   unset ORACLE_HOME

   unset ORACLE_SID

2. Create the Oracle DB home directory and change the directory to it.

   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.

   unzip -q /tmp/archive/LINUX.X64_193000_db_home.zip

4. From the DB home, delete the OPatch directory.

   rm -rf OPatch

5. From DB home, unzip p6880880_190000_Linux-x86-64.zip.

   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.

   vi cv/admin/cvu_config

7. From the /tmp/archive directory, unpack the DB 19.18 RU patch.

   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
---------- ----------  ------------   ------------------------
----------  -----------  -----------   ---------------
```

144
<table>
<thead>
<tr>
<th>Resource</th>
<th>Status</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora.DA.TA.dg</td>
<td>ONLINE</td>
<td>ora_02, STABLE</td>
</tr>
<tr>
<td>ora.LISTENER.lsnr</td>
<td>ONLINE</td>
<td>ora_02, Intermediate, Not All Endpoints Registered, STABLE</td>
</tr>
<tr>
<td>ora.LOGS.dg</td>
<td>ONLINE</td>
<td>ora_02, STABLE</td>
</tr>
<tr>
<td>ora.asm</td>
<td>ONLINE</td>
<td>ora_02, Started, STABLE</td>
</tr>
<tr>
<td>ora.ons</td>
<td>OFFLINE</td>
<td>ora_02, STABLE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>Status</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>ora.cdb3.db</td>
<td>1 ONLINE</td>
<td>ora_02, Open, HOME=/u01/app/oracle/product/19.0.0, STABLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/cdb3, STABLE</td>
</tr>
<tr>
<td>ora.cdb4.db</td>
<td>1 ONLINE</td>
<td>ora_02, Open, HOME=/u01/app/oracle/product/19.0.0, STABLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/cdb4, STABLE</td>
</tr>
<tr>
<td>ora.cssd</td>
<td>1 ONLINE</td>
<td>ora_02, STABLE</td>
</tr>
<tr>
<td>ora.diskmon</td>
<td>OFFLINE</td>
<td>ora_02, STABLE</td>
</tr>
<tr>
<td>ora.driverafd</td>
<td>ONLINE</td>
<td>ora_02, STABLE</td>
</tr>
<tr>
<td>ora.evmd</td>
<td>ONLINE</td>
<td>ora_02, STABLE</td>
</tr>
</tbody>
</table>
3. Set the Oracle user .bash_profile.

```bash
vi ~/.bash_profile
```

Add following entries:

```bash
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.

```bash
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

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> 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/undotbs1.274.1149422017
+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
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;
alter database archivelog;

alter database open;

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

#### Primary Backup(s)

<table>
<thead>
<tr>
<th>Backup Name</th>
<th>Count</th>
<th>Type</th>
<th>Size</th>
<th>Verified</th>
<th>Mounted</th>
<th>HWN Cataloged</th>
<th>SCN</th>
</tr>
</thead>
<tbody>
<tr>
<td>one_2_1601_20230403_014009_1</td>
<td>1</td>
<td>Log</td>
<td>181/2GB</td>
<td>Not Applicable</td>
<td>False</td>
<td>Yes</td>
<td>2000000</td>
</tr>
<tr>
<td>one_2_1601_20230403_014009_2</td>
<td>1</td>
<td>Log</td>
<td>181/2GB</td>
<td>Not Applicable</td>
<td>False</td>
<td>Yes</td>
<td>2000000</td>
</tr>
<tr>
<td>one_2_1601_20230403_014009_3</td>
<td>1</td>
<td>Log</td>
<td>181/2GB</td>
<td>Not Applicable</td>
<td>False</td>
<td>Yes</td>
<td>2000000</td>
</tr>
<tr>
<td>one_2_1601_20230403_014009_4</td>
<td>1</td>
<td>Log</td>
<td>181/2GB</td>
<td>Not Applicable</td>
<td>False</td>
<td>Yes</td>
<td>2000000</td>
</tr>
</tbody>
</table>

- **Backup Name**: The name of each backup file.
- **Count**: The number of backups.
- **Type**: Log or Data.
- **Size**: The size of each backup file.
- **Verified**: Whether the backup is verified or not.
- **Mounted**: Whether the backup is mounted or not.
- **HWN Cataloged**: Whether the backup is cataloged in the HWN.
- **SCN**: The SCN (Sequence Command Number) of the backup.
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.

```
[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.
```
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.

```
[oracle@ora_02 bin]$ ./crsctl stat res -t

Name          Target State   Server          State details
---------------------------------------------------------------
ora.DATA.dg    ONLINE ONLINE   ora_02           STABLE
ora.LISTENER.1snr ONLINE INTERMEDIATE ora_02 Not All Endpoints Re
gistered,STABLE
ora.LOG3.dg     ONLINE ONLINE   ora_02           STABLE
ora.LOG3_CDB3_22_dg ONLINE ONLINE   ora_02           STABLE
ora.asm        ONLINE ONLINE   ora_02           Started,STABLE
ora.ons         OFFLINE OFFLINE ora_02           STABLE

Cluster Resources
ora.cdb3.db 1 ONLINE INTERMEDIATE ora_02 Dismounted, Mount Ini
tiated, HOME=/u01/app/oracle/product/19.0.0/cdb3, STABLE
ora.cdb4.db 1 ONLINE ONLINE   ora_02 Open, HOME=/u01/app/oracle/product/19.0.0/cdb4, STABLE
ora.csdd      1 ONLINE ONLINE   ora_02           STABLE
ora.dismon    1 OFFLINE OFFLINE           STABLE
ora.driver.efd 1 ONLINE ONLINE   ora_02           STABLE
ora.evmd      1 ONLINE ONLINE   ora_02           STABLE
```
12. From DB VM ora_02, validate the dropped table is recovered after a successful recovery.

```bash
[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:
```
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>
<th>DT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>06-OCT-23 03.18.24.000000 PM</td>
</tr>
<tr>
<td></td>
<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 05.02.20.382702 PM -07:00</td>
</tr>
</tbody>
</table>

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.lsnr
   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
Started,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/cdb3,STABLE
/oracle/product/19.0.0
ora.cdb3tst.db
   1 ONLINE ONLINE  ora_02
Open,HOME=/u01/app/oracle/product/19.0.0/cdb3tst,STABLE
```
Oracle/product/19.0.0

/cdb4,STABLE
ora.cdb4.db
  1  ONLINE  ONLINE  ora_02
Open,HOME=/u01/app/oracle/product/19.0.0/cdb4

/oracle@ora_02 ~]$ export
ORACLE_HOME=/u01/app/oracle/product/19.0.0/cdb4
/oracle@ora_02 ~]$ export ORACLE_SID=cdb3tst
/oracle@ora_02 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Sat Oct 7 08:04:51 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
-------- --------------------
CDB3TST  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> alter session set container=CDB3_PDB1;

Session altered.

SQL> select * from test;

<table>
<thead>
<tr>
<th>ID</th>
<th>DT</th>
<th>EVENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>06-OCT-23 03.18.24.000000 PM</td>
<td>test oracle recovery on guest mounted fsx storage to VMC guest vm ora_02</td>
</tr>
</tbody>
</table>

SQL>

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-
TR-4981: Oracle Active Data Guard Cost Reduction with Amazon FSx ONTAP

Allen Cao, Niyaz Mohamed, NetApp

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
might not match the final deployment environment. For more information, see the section [Key Factors for Deployment Consideration].

**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>Description</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>Specification</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>
</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, switchover, and failover 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 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 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.

```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.

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.

```bash
[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

--
Name      Target  State       Server    State details
---------  --------  ---------  --------  ------------------------------------------
---------  --------  ---------  --------  ------------------------------------------
Local Resources
```
ora.DATA.dg
ONLINE  ONLINE       ip-172-30-15-45          STABLE
ora.LISTENER.lsnr
ONLINE  ONLINE       ip-172-30-15-45          STABLE
ora.LOGS.dg
ONLINE  ONLINE       ip-172-30-15-45          STABLE
ora.asm
ONLINE  ONLINE       ip-172-30-15-45
Started,STABLE
ora.ons
OFFLINE OFFLINE       ip-172-30-15-45          STABLE

----------------------------------------
Cluster Resources
----------------------------------------
ora.cssd
  1        ONLINE  ONLINE       ip-172-30-15-45          STABLE
ora.db1.db
  1        ONLINE  ONLINE       ip-172-30-15-45
Open,HOME=/u01/app/oracle/product/19.0.0/db1,STABLE
ora.diskmon
  1        OFFLINE OFFLINE                               STABLE
ora.driver.afd
  1        ONLINE  ONLINE       ip-172-30-15-45          STABLE
ora.evmd
  1        ONLINE  ONLINE       ip-172-30-15-45          STABLE

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.

```
alter database add standby logfile thread 1 size 200M;
```

Validate after standby logs addition:

```
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.

```
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.
# Backup file is /u01/app/oracle/crsdata/ip-172-30-15-45/output/listener.ora.bak.ip-172-30-15-45.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-172-30-15-45.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.

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.

```bash
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.

   export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db1

   export PATH=$PATH:$ORACLE_HOME/bin

7. Use dbca to instantiate standby database from primary database db1.

```sql
[oracle@ip-172-30-15-67 bin]$ export ORACLE_SID=db1
[oracle@ip-172-30-15-67 bin]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Aug 30 18:25:46 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;

<table>
<thead>
<tr>
<th>NAME</th>
<th>OPEN_MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB1</td>
<td>READ ONLY</td>
</tr>
</tbody>
</table>
```
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></td>
<td></td>
</tr>
</tbody>
</table>

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></td>
</tr>
<tr>
<td>DG_CONFIG=(db1_NY,db1_LA)</td>
<td></td>
<td></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>
<tr>
<td>NAME</td>
</tr>
<tr>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>+DATA/DB1_LA/03C5C01A66EE9797E0632D0F1EAC5F59/DATAFILE/sysaux.265.11</td>
</tr>
<tr>
<td>+DATA/DB1_LA/DATAFILE/users.266.1146248267</td>
</tr>
<tr>
<td>+DATA/DB1_LA/03C5C01A66EE9797E0632D0F1EAC5F59/DATAFILE/undotbs1.267.1146248269</td>
</tr>
<tr>
<td>+DATA/DB1_LA/03C5EFD07C41A1FAE0632D0F1EAC9BD8/DATAFILE/system.268.1146248271</td>
</tr>
<tr>
<td>+DATA/DB1_LA/03C5EFD07C41A1FAE0632D0F1EAC9BD8/DATAFILE/sysaux.269.1146248279</td>
</tr>
<tr>
<td>+DATA/DB1_LA/03C5EFD07C41A1FAE0632D0F1EAC9BD8/DATAFILE/undotbs1.270.1146248285</td>
</tr>
<tr>
<td>+DATA/DB1_LA/03C5EFD07C41A1FAE0632D0F1EAC9BD8/DATAFILE/users.271.1146248293</td>
</tr>
</tbody>
</table>

19 rows selected.

SQL> select name from v$controlfile;

<table>
<thead>
<tr>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>+DATA/DB1_LA/03C5F0DDF35CA2B6E0632D0F1EAC8B6B/DATAFILE/system.272.1146248295</td>
</tr>
<tr>
<td>+DATA/DB1_LA/03C5F0DDF35CA2B6E0632D0F1EAC8B6B/DATAFILE/sysaux.273.1146248301</td>
</tr>
<tr>
<td>+DATA/DB1_LA/03C5F0DDF35CA2B6E0632D0F1EAC8B6B/DATAFILE/undotbs1.274.1146248309</td>
</tr>
<tr>
<td>+DATA/DB1_LA/03C5F0DDF35CA2B6E0632D0F1EAC8B6B/DATAFILE/users.275.1146248315</td>
</tr>
<tr>
<td>+DATA/DB1_LA/03C5F1C9B142A2F1E0632D0F1EACF21A/DATAFILE/system.276.1146248323</td>
</tr>
<tr>
<td>+DATA/DB1_LA/03C5F1C9B142A2F1E0632D0F1EACF21A/DATAFILE/sysaux.277.1146248331</td>
</tr>
<tr>
<td>+DATA/DB1_LA/03C5F1C9B142A2F1E0632D0F1EACF21A/DATAFILE/undotbs1.278.1146248347</td>
</tr>
<tr>
<td>+DATA/DB1_LA/03C5F1C9B142A2F1E0632D0F1EACF21A/DATAFILE/users.279.1146248353</td>
</tr>
</tbody>
</table>

19 rows selected.

SQL> select name from v$tempfile;

<table>
<thead>
<tr>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>+DATA/DB1_LA/CONTROLFILE/current.260.1146248209</td>
</tr>
<tr>
<td>+LOGS/DB1_LA/CONTROLFILE/current.257.1146248209</td>
</tr>
</tbody>
</table>
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.
SQL> SELECT ROLE, THREAD#, SEQUENCE#, ACTION FROM V$DATAGUARD_PROCESS;

<table>
<thead>
<tr>
<th>ROLE</th>
<th>THREAD#</th>
<th>SEQUENCE#</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>recovery apply slave</td>
<td>0</td>
<td>0</td>
<td>IDLE</td>
</tr>
<tr>
<td>recovery apply slave</td>
<td>0</td>
<td>0</td>
<td>IDLE</td>
</tr>
<tr>
<td>recovery apply slave</td>
<td>0</td>
<td>0</td>
<td>IDLE</td>
</tr>
<tr>
<td>recovery apply slave</td>
<td>0</td>
<td>0</td>
<td>IDLE</td>
</tr>
<tr>
<td>recovery logmerger</td>
<td>1</td>
<td>30</td>
<td>APPLYING_LOG</td>
</tr>
<tr>
<td>RFS ping</td>
<td>1</td>
<td>30</td>
<td>IDLE</td>
</tr>
<tr>
<td>RFS async</td>
<td>1</td>
<td>30</td>
<td>IDLE</td>
</tr>
<tr>
<td>archive redo</td>
<td>0</td>
<td>0</td>
<td>IDLE</td>
</tr>
<tr>
<td>archive redo</td>
<td>0</td>
<td>0</td>
<td>IDLE</td>
</tr>
<tr>
<td>archive redo</td>
<td>0</td>
<td>0</td>
<td>IDLE</td>
</tr>
<tr>
<td>gap manager</td>
<td>0</td>
<td>0</td>
<td>IDLE</td>
</tr>
<tr>
<td>managed recovery</td>
<td>0</td>
<td>0</td>
<td>IDLE</td>
</tr>
<tr>
<td>redo transport monitor</td>
<td>0</td>
<td>0</td>
<td>IDLE</td>
</tr>
<tr>
<td>log writer</td>
<td>0</td>
<td>0</td>
<td>IDLE</td>
</tr>
<tr>
<td>archive local</td>
<td>0</td>
<td>0</td>
<td>IDLE</td>
</tr>
<tr>
<td>redo transport timer</td>
<td>0</td>
<td>0</td>
<td>IDLE</td>
</tr>
</tbody>
</table>

16 rows selected.

SQL>

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.

```
alter system set dg_broker_start=true scope=both;
```

2. From primary database, connect to Data Guard Borker 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 relevant 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.

```
[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.

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.

![NetApp SnapCenter Screenshot]

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.
15. 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.
[oracle@ip-172-30-15-126 ~]$ export ORACLE_HOME=/u01/app/oracle/product/19.0.0/dev
[oracle@ip-172-30-15-126 ~]$ export ORACLE_SID=db1dev
[oracle@ip-172-30-15-126 ~]$ export PATH=$PATH:$ORACLE_HOME/bin
[oracle@ip-172-30-15-126 ~]$ sqlplus / as sysdba

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;

<table>
<thead>
<tr>
<th>ID</th>
<th>DT</th>
<th>EVENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>215</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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-8225-4784bed9843&sc_channel=ps&s_kwcid=AL4422!3!467723097970!e!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARIsAjdSrdqwQrghn6l71jiWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:G:s&s_kwcid=AL4422!3!467723097970!e!!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>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSx ONTAP storage</td>
<td>Current version offered by AWS</td>
</tr>
<tr>
<td>EC2 instance for compute</td>
<td>t2.xlarge/4vCPU/16G</td>
</tr>
<tr>
<td></td>
<td>One FSx HA cluster in the same VPC and availability zone</td>
</tr>
<tr>
<td></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>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>
</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 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.

- **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.

   ```bash
   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.

   ```bash
   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.

   ```
   mkdir /nfsfsxn/oracopy
   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.

   ```
   sqlplus / as sysdba
   From sqlplus prompt, execute following command.
   ```

   ```
   alter database enable block change tracking using file '/nfsfsxn/oracopy/bct_db1.ctf'
   ```

   ```
   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.

```bash
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.

```bash
RMAN> @/home/oracle/rman_bkup_merge.cmd
```
RMAN> RUN
2> { 
3>   allocate channel c1 device type disk format '/nfsfsxn/oracopy/%U';
4>   allocate channel c2 device type disk format '/nfsfsxn/oracopy/%U';
5>   allocate channel c3 device type disk format '/nfsfsxn/oracopy/%U';
6>   allocate channel c4 device type disk format '/nfsfsxn/oracopy/%U';
7>   delete obsolete;
8>   sql 'alter system archive log current';
9>   recover copy of database with tag 'OraCopyBKUPonFSxN_level_0';
10>  backup incremental level 1 copies=1 for recover of copy with tag 'OraCopyBKUPonFSxN_level_0' database;
11> }

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.291.113
7018481
input datafile file number=00030
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.295.113
7018787
input datafile file number=00011
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/undotbs1.27
1.1136668041
input datafile file number=00035
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.300.113
7019181
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.288.113
7018359
input datafile file number=00027
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.292.113
7018523
input datafile file number=00031
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.296.113
7018837
input datafile file number=00009
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/system.27.1136668041
input datafile file number=00034
name=+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.299.113
7019117
.
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</th>
<th>Status</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</td>
<td>17-MAY-23</td>
<td>3009819</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<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></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>A</td>
<td>17-MAY-23</td>
<td>3009826</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<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></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>4</td>
<td>A</td>
<td>17-MAY-23</td>
<td>3009830</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<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></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>5</td>
<td>A</td>
<td>17-MAY-23</td>
<td>2383520</td>
<td>12-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<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></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Container ID: 2, PDB Name: PDB$SEED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>6</td>
<td>A</td>
<td>17-MAY-23</td>
<td>2383520</td>
<td>12-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-6_0o1sd7c8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Container ID: 2, PDB Name: PDB$SEED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>7</td>
<td>A</td>
<td>17-MAY-23</td>
<td>3009907</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<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></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>8</td>
<td>A</td>
<td>17-MAY-23</td>
<td>2383520</td>
<td>12-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<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></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>Container ID</th>
<th>PDB Name</th>
<th>Date</th>
<th>Time</th>
<th>Size</th>
<th>Date</th>
<th>Time</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>9</td>
<td>A 17-MAY-23</td>
<td>3009871</td>
<td>17-MAY-23</td>
<td>NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-9_0q1sd7cm</td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID: 3, PDB Name: DB1_PDB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>10</td>
<td>A 17-MAY-23</td>
<td>3009849</td>
<td>17-MAY-23</td>
<td>NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-10_0k1sd7bb</td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID: 3, PDB Name: DB1_PDB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>11</td>
<td>A 17-MAY-23</td>
<td>3009862</td>
<td>17-MAY-23</td>
<td>NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-11_0n1sd7c1</td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID: 3, PDB Name: DB1_PDB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>12</td>
<td>A 17-MAY-23</td>
<td>3009909</td>
<td>17-MAY-23</td>
<td>NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-12_111sd7dm</td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID: 3, PDB Name: DB1_PDB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>13</td>
<td>A 17-MAY-23</td>
<td>3009876</td>
<td>17-MAY-23</td>
<td>NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-13_0rlsd7ct</td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID: 4, PDB Name: DB1_PDB2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>14</td>
<td>A 17-MAY-23</td>
<td>3009854</td>
<td>17-MAY-23</td>
<td>NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-14_011sd7bi</td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID: 4, PDB Name: DB1_PDB2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>15</td>
<td>A 17-MAY-23</td>
<td>3009900</td>
<td>17-MAY-23</td>
<td>NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-15_0t1sd7db</td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID: 4, PDB Name: DB1_PDB2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>16</td>
<td>A 17-MAY-23</td>
<td>3009911</td>
<td>17-MAY-23</td>
<td>NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-12_111sd7dm</td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container ID: 4, PDB Name: DB1_PDB2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No.</td>
<td>PDB No.</td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td>Container ID</td>
<td>PDB Name:</td>
<td>Size</td>
<td>Date</td>
<td>Status</td>
</tr>
<tr>
<td>-----</td>
<td>---------</td>
<td>--------------------------------</td>
<td>--------------</td>
<td>----------</td>
<td>------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>30</td>
<td>17</td>
<td>17-MAY-23</td>
<td></td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-17_0s1sd7d4</td>
<td>3009895</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>17-MAY-23</td>
<td></td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-18_0mlsd7bq</td>
<td>3009858</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td>32</td>
<td>19</td>
<td>17-MAY-23</td>
<td></td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-19_0u1sd7de</td>
<td>3009903</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td>37</td>
<td>20</td>
<td>17-MAY-23</td>
<td></td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-20_131sd7do</td>
<td>3009914</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td>4</td>
<td>21</td>
<td>17-MAY-23</td>
<td></td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-21_021sd6pv</td>
<td>3009019</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td>5</td>
<td>22</td>
<td>17-MAY-23</td>
<td></td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-22_031sd6r2</td>
<td>3009419</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>17-MAY-23</td>
<td></td>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-23_041sd6s5</td>
<td>3009460</td>
<td>17-MAY-23</td>
<td>NO</td>
</tr>
</tbody>
</table>
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
## 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 TEMP</td>
<td>32767</td>
<td>+DATA/DB1/TEMPFILE/temp.265.1136666447</td>
</tr>
<tr>
<td>2</td>
<td>123 PDB$SEED:TEMP</td>
<td>32767</td>
<td>+DATA/DB1/FB864A929AEB79B9E053630F1EAC7046/TEMPFILE/temp.269.1136667</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>185</td>
</tr>
<tr>
<td>3</td>
<td>10240 DB1_PDB1:TEMP</td>
<td>32767</td>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/TEMPFILE/temp.274.1136668</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>051</td>
</tr>
<tr>
<td>4</td>
<td>123 DB1_PDB2:TEMP</td>
<td>32767</td>
<td>+DATA/DB1/FB867EA89ECF81C0E053630F1EACB901/TEMPFILE/temp.279.1136668</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>067</td>
</tr>
<tr>
<td>5</td>
<td>123 DB1_PDB3:TEMP</td>
<td>32767</td>
<td>+DATA/DB1/FB867F8A4D4F821CE053630F1EAC69CC/TEMPFILE/temp.284.1136668</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>081</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_031sd6t2
-rw-r----- 1 oracle asm 4294975488 May 17 18:10 data_D-DB1_I-
1730530050_TS-SOE_FNO-23_041sd6s5
-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-
1730530050_TS-SOE_FNO-30_0b1sd73n
```

235
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.

```
[ec2-user@ip-172-30-15-99 ~]$ 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 sysdba

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 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> 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>18-MAY-23 02.35.37.000000 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 1.2885E+10 bytes
Fixed Size 9177880 bytes
Variable Size 1778384896 bytes
Database Buffers 1.1073E+10 bytes
Redo Buffers 24375296 bytes
Database mounted.

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_0psd7cf"
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_0q1sdcm
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_D-
DB1_I-1730530050_TS-SYSAUX_FNO-10_0k1sd7bb
destination for restore of datafile 00021: /nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-21_021sd6pv
destination for restore of datafile 00025: /nfsfsxn/oracopy/data_D-
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_D-
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_D-
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> 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;
```

<table>
<thead>
<tr>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>+DATA/DB1/CONTROLFILE/current.261.1136666435</td>
</tr>
<tr>
<td>+DATA/DB1/FB864A929AEB79B9E053630F1EAC7046/TEMPFILE/temp.269.1136667185</td>
</tr>
<tr>
<td>+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/TEMPFILE/temp.274.1136668051</td>
</tr>
<tr>
<td>+DATA/DB1/FB867EA89ECF81C0E053630F1EACB901/TEMPFILE/temp.279.1136668067</td>
</tr>
<tr>
<td>+DATA/DB1/FB867F8A4D4F821CE053630F1EAC69CC/TEMPFILE/temp.284.1136668081</td>
</tr>
<tr>
<td>+DATA/DB1/ONLINELOG/group_1.262.1136666437</td>
</tr>
<tr>
<td>+DATA/DB1/ONLINELOG/group_2.263.1136666437</td>
</tr>
<tr>
<td>+DATA/DB1/ONLINELOG/group_3.264.1136666437</td>
</tr>
<tr>
<td>+DATA/DB1/TEMPFILE/temp.265.1136666447</td>
</tr>
<tr>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-21_021sd6pv</td>
</tr>
<tr>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-22_031sd6r2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-23_041sd6s5</td>
</tr>
<tr>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-24_051sd6t9</td>
</tr>
<tr>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-25_061sd6uc</td>
</tr>
<tr>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-26_071sd6vf</td>
</tr>
<tr>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-27_081sd70i</td>
</tr>
<tr>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-28_091sd71l</td>
</tr>
<tr>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-29_0a1sd72o</td>
</tr>
<tr>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-30_0b1sd73r</td>
</tr>
<tr>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-31_0c1sd74u</td>
</tr>
<tr>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-32_0d1sd762</td>
</tr>
<tr>
<td>/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-33_0e1sd775</td>
</tr>
</tbody>
</table>
```
6. From SQL plus, check the content of test table we have inserted before the switch over to copy.
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>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
</tr>
<tr>
<td>EVENT</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

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.

```
[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
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

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;

246
```
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 kernel 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, orainst.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,noint
```

7. Validate the Oracle database backup image copy on FSxN NFS mount point.

```
[ec2-user@ip-172-30-15-124 ~]$ ls -ltr /nfsfsxn/oracopy
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 6l1tf6bb_213_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 6g1tf6bb_208_1_1
-rw-r----- 1 oracle 54331 589824 May 30 15:29 6f1tf6bb_213_1_1
-rw-r----- 1 oracle 54331 606208 May 30 15:29 6m1tf6bb_214_1_1
-rw-r----- 1 oracle 54331 368640 May 30 15:29 6o1tf6bb_216_1_1
-rw-r----- 1 oracle 54331 368640 May 30 15:29 6p1tf6bc_217_1_1
-rw-r----- 1 oracle 54331 57344 May 30 15:29 6r1tf6bc_219_1_1
-rw-r----- 1 oracle 54331 57344 May 30 15:29 6s1tf6bc_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 sequence number. In this case, it is 175. Our recovery point is up to log sequence number 176.
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.

```
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 Completion Time</th>
<th>Ckp SCN</th>
<th>Ckp Time</th>
<th>Sparse</th>
</tr>
</thead>
<tbody>
<tr>
<td>316</td>
<td>30-MAY-23 4120170</td>
<td>30-MAY-23</td>
<td>NO</td>
<td></td>
</tr>
</tbody>
</table>
|     | Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
   | SYSTEM_FNO-1_4f1t506m |
|     | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
| 322 | 30-MAY-23 4120175     | 30-MAY-23 | NO |
|     | Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
   | SYSAUX_FNO-3_4g1t506m |
|     | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
| 317 | 30-MAY-23 4120179     | 30-MAY-23 | NO |
|     | Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
   | UNDOTBS1_FNO-4_4h1t5083 |
|     | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
| 221 | 26-MAY-23 2383520     | 12-MAY-23 | NO |
|     | Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
   | SYSTEM_FNO-5_4q1t509n |
|     | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
|     | Container ID: 2, PDB Name: PDB$SEED |
| 216 | 26-MAY-23 2383520     | 12-MAY-23 | NO |
|     | Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
   | SYSAUX_FNO-6_4m1t508t |
|     | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
|     | Container ID: 2, PDB Name: PDB$SEED |
| 323 | 30-MAY-23 4120207     | 30-MAY-23 | NO |
|     | Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
   | USERS_FNO-7_4u1t50a6 |
|     | Tag: ORACOPYBKUPONFSXN_LEVEL_0 |
| 227 | 26-MAY-23 2383520     | 12-MAY-23 | NO |
|     | Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
<p>| UNDOTBS1_FNO-8_4t1t50a6 |</p>
<table>
<thead>
<tr>
<th>Container ID</th>
<th>PDB Name</th>
<th>Tag: ORACOPYBKUPONFSXN_LEVEL_0</th>
<th>A</th>
<th>30-MAY-23</th>
<th>4120158</th>
<th>30-MAY-23</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PDB$SEED</td>
<td></td>
<td>9</td>
<td>30-MAY-23</td>
<td></td>
<td>30-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-9_4n1t509m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tag: ORACOPYBKUPONFSXN_LEVEL_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Container ID: 2, PDB Name: PDB$SEED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>DB1_PDB1</td>
<td></td>
<td>10</td>
<td>30-MAY-23</td>
<td>4120166</td>
<td>30-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-10_4i1t5083</td>
<td></td>
<td></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>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Container ID: 3, PDB Name: DB1_PDB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>DB1_PDB2</td>
<td></td>
<td>11</td>
<td>30-MAY-23</td>
<td>4120154</td>
<td>30-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-11_4l1t508t</td>
<td></td>
<td></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>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Container ID: 3, PDB Name: DB1_PDB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>DB1_PDB1</td>
<td></td>
<td>12</td>
<td>30-MAY-23</td>
<td>4120162</td>
<td>30-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-12_4v1t50aa</td>
<td></td>
<td></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>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Container ID: 3, PDB Name: DB1_PDB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>DB1_PDB2</td>
<td></td>
<td>13</td>
<td>30-MAY-23</td>
<td>4120191</td>
<td>30-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-13_4o1t509m</td>
<td></td>
<td></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>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Container ID: 4, PDB Name: DB1_PDB2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>DB1_PDB2</td>
<td></td>
<td>14</td>
<td>30-MAY-23</td>
<td>4120183</td>
<td>30-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-14_4j1t508s</td>
<td></td>
<td></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>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Container ID: 4, PDB Name: DB1_PDB2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>DB1_PDB2</td>
<td></td>
<td>15</td>
<td>30-MAY-23</td>
<td>4120199</td>
<td>30-MAY-23</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-15_4r1t50a6</td>
<td></td>
<td></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>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Container ID: 4, PDB Name: DB1_PDB2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>DB1_PDB2</td>
<td></td>
<td>16</td>
<td>30-MAY-23</td>
<td>4120211</td>
<td>30-MAY-23</td>
<td>NO</td>
</tr>
</tbody>
</table>
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-SOE_FNO-20_511t50ad
  Tag: ORACOPYBKUPONFSXN_LEVEL_0
  Container ID: 3, PDB Name: DB1_PDB1

298  21 A 30-MAY-23  4120166  30-MAY-23 NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-21_301t4ut2
  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
A 30-MAY-23 4120162 30-MAY-23 NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-24_3r1t4ut3
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

A 30-MAY-23 4120166 30-MAY-23 NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-25_3s1t4v1a
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

A 30-MAY-23 4120154 30-MAY-23 NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-26_451t4vt7
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

A 30-MAY-23 4120158 30-MAY-23 NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-27_461t4vt7
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

A 30-MAY-23 4120162 30-MAY-23 NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-28_471t4vt7
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

A 30-MAY-23 4120166 30-MAY-23 NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-29_481t4vt7
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

A 30-MAY-23 4120154 30-MAY-23 NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-30_491t5014
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

A 30-MAY-23 4120158 30-MAY-23 NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-31_4a1t5015
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

312 32 A 30-MAY-23 4120162 30-MAY-23 NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-32_4b1t501u
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

314 33 A 30-MAY-23 4120162 30-MAY-23 NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-33_4c1t501v
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

304 34 A 30-MAY-23 4120158 30-MAY-23 NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-34_4d1t5058
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

311 35 A 30-MAY-23 4120154 30-MAY-23 NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-35_4e1t5059
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

16. Switch database to copy to run recovery without database restore.

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
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- USERS_FNO-7_4u1t50a6"
datafile 8 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-8_4t1t50a6"
datafile 9 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_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-UNDOTBS1_FNO-11_4l1t508t"
datafile 12 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS- USERS_FNO-12_4v1t50a6"
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-UNDOTBS1_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-UNDOTBS1_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-
Datafile 23 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-23_3q1t4ut3"
datafile 24 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-24_3r1t4ut3"
datafile 25 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-25_3s1t4v1a"
datafile 26 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-26_451t4vt7"
datafile 27 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-27_461t4vt7"
datafile 28 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-28_471t4vt7"
datafile 29 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-29_481t4vt7"
datafile 30 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-30_491t5014"
datafile 31 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-31_4a1t5015"
datafile 32 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-32_4b1t501u"
datafile 33 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-33_4c1t501v"
datafile 34 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-34_4d1t5058"
datafile 35 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-35_4e1t5059"

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> }

executing 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_.ar

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_.ar

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_.ar

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_.ar

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_.ar

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_.ar

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__03mfxl7v_.ar

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_.ar

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__03wgdry_.ar

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_.ar

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_.ar

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_.ar
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_.ar
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_.ar
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_.ar
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_.ar
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_.ar
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_.ar
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_.ar
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__05h81mlh_.ar
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_.ar
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_.ar
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_.arc

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_.arc

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_.arc

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_.arc

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__060kgh33_.arc

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_.arc

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_.arc

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__067wnwy8_.arc

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_.arc

archived log for thread 1 with sequence 175 is already on disk as file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_175__08c7jc2b_.arc

archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_142__02n3x2qb_.arc thread=1 sequence=142

archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_143__02rotwy
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__05krsqmhm_.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__05s3o01r_.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__05y45qd3d_.arc thread=1 sequence=169
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_170__060kgv33_.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
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_174__06b9zd8_.arc thread=1 sequence=174
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.
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 test oracle incremental merge switch to copy</td>
</tr>
<tr>
<td>2</td>
<td>30-MAY-23 05.23.11.000000 PM test recovery on a new EC2 instance host with image copy on FSxN</td>
</tr>
</tbody>
</table>

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
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;
```

<table>
<thead>
<tr>
<th>ID</th>
<th>DT</th>
<th>EVENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>18-MAY-23 02.35.37.000000 PM</td>
<td>test oracle incremental merge switch to copy</td>
</tr>
<tr>
<td>2</td>
<td>30-MAY-23 05.23.11.000000 PM</td>
<td>test recovery on a new EC2 instance host with image copy on FSxN</td>
</tr>
<tr>
<td>3</td>
<td>05-JUN-23 03.19.46.000000 PM</td>
<td>test clone on a new EC2 instance host with image copy on FSxN</td>
</tr>
</tbody>
</table>

```sql
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
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.
```plaintext
FsxId06c3c8b2a7bd56458::> vol snapshot create -vserver svm_ora -volume ora_01_copy -snapshot one-off.2023-06-05-1137 -foreground true

FsxId06c3c8b2a7bd56458::> snapshot show

---Blocks---
Vserver  Volume   Snapshot                                  Size
Total% Used%
-------- -------- ------------------------------------- --------
------ ----- -----
svm_ora  ora_01_copy
daily.2023-06-02_0010                   3.59GB
 2%    5%
daily.2023-06-03_0010                   1.10GB
 1%    1%
daily.2023-06-04_0010                    608KB
 0%    0%
daily.2023-06-05_0010                   3.81GB
 2%    5%
one-off.2023-06-05-1137                  168KB
 0%    0%

svm_ora_root
weekly.2023-05-28_0015                  1.86MB
 0%   78%
daily.2023-06-04_0010                    152KB
 0%   22%
weekly.2023-06-04_0015                  1.24MB
 0%   70%
daily.2023-06-05_0010                    196KB
 0%   27%
hourly.2023-06-05_1005                   156KB
 0%   22%
hourly.2023-06-05_1105                   156KB
 0%   22%
hourly.2023-06-05_1205                   156KB
 0%   22%
hourly.2023-06-05_1305                   156KB
 0%   22%
hourly.2023-06-05_1405                  1.87MB
 0%   78%
hourly.2023-06-05_1505                   148KB
 0%   22%
15 entries were displayed.
```
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.

```
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 -parent-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%
--------- ------------ ------------ ---------- ---- ----------
---------- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
7. Validate the database incremental backup sets, image copy, and available archived logs on FSx NFS mount.

```
    total 79450332
-rw-r----- 1 oracle 54331 482353152 Jun  1 19:02 data_D-DB1_I-
    1730530050_TS-SYSAUX_FNO-6_891tkhr
-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 8s1tl05t_284_1_1
-rw-r----- 1 oracle 54331  3514368 Jun  1 20:21 8t1tl05t_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 20:31 901tl062_287_1_1
-rw-r----- 1 oracle 54331  245760 Jun  1 20:31 911tl066_289_1_1
-rw-r----- 1 oracle 54331  237568 Jun  1 20:31 931tl064_290_1_1
-rw-r----- 1 oracle 54331  257584 Jun  1 20:31 941tl065_291_1_1
-rw-r----- 1 oracle 54331  1040760832 Jun  1 20:31 961tl067_292_1_1
-rw-r----- 1 oracle 54331  344064 Jun  1 20:31 971tl065_293_1_1
-rw-r----- 1 oracle 54331  534781952 Jun  1 20:31 981tl067_294_1_1
-rw-r----- 1 oracle 54331  534781952 Jun  1 20:31 991tl068_295_1_1
-rw-r----- 1 oracle 54331  534781952 Jun  1 20:31 9a1tl068_296_1_1
-rw-r----- 1 oracle 54331  534781952 Jun  1 20:31 9b1tl069_297_1_1
-rw-r----- 1 oracle 54331  534781952 Jun  1 20:31 9c1tl070_298_1_1
-rw-r----- 1 oracle 54331  534781952 Jun  1 20:31 9d1tl071_299_1_1
-rw-r----- 1 oracle 54331  534781952 Jun  1 20:31 9e1tl072_300_1_1
-rw-r----- 1 oracle 54331 1121984512 Jun  5 15:21 data_D-DB1_I-
    1730530050_TS-SYSTEM_FNO-1_821tkrb8
-rw-r----- 1 oracle 54331 1027612672 Jun  5 15:21 data_D-DB1_I-
    1730530050_TS-SYSAUX_FNO-3_831tkrd9
-rw-r----- 1 oracle 54331  429924352 Jun  5 15:21 data_D-DB1_I-
    1730530050_TS-SYSTEM_FNO-9_8altkrhr
-rw-r----- 1 oracle 54331  707796992 Jun  5 15:21 data_D-DB1_I-
    1730530050_TS-UNDOTBS1_FNO-4_851tkrgf
-rw-r----- 1 oracle 54331  534781952 Jun  5 15:21 data_D-DB1_I-
    1730530050_TS-SYSAUX_FNO-14_871tkhr
-rw-r----- 1 oracle 54331  534781952 Jun  5 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: AUTORECOVERY

/nfsfsxn/archlog/DB1/autobackup/2023_06_05/o1_mf_s_1138721482_l7vzybyq_.bkp found in the recovery area
channel ORA_DISK_1: looking for AUTORECOVERY on day: 20230607
channel ORA_DISK_1: looking for AUTORECOVERY on day: 20230606
channel ORA_DISK_1: looking for AUTORECOVERY on day: 20230605
channel ORA_DISK_1: restoring control file from AUTORECOVERY

/nfsfsxn/archlog/DB1/autobackup/2023_06_05/o1_mf_s_1138721482_l7vzybyq_.bkp
channel ORA_DISK_1: control file restore from AUTORECOVERY 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
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSAUX_FNO-3_831tkrd9
  Tag: ORACOPYBKUPONFSXN_LEVEL_0

365 4 A 05-JUN-23 8319171 01-JUN-23 NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-4_851tkrjg
  Tag: ORACOPYBKUPONFSXN_LEVEL_0

355 5 A 01-JUN-23 2383520 12-MAY-23 NO
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSTEM_FNO-5_8d1tkril
  Tag: ORACOPYBKUPONFSXN_LEVEL_0
  Container ID: 2, PDB Name: PDB$SEED

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_8hltkrj9
  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_8altkrhr
  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_86ltkrho
  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-
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_8e1tkrja
    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_8cltkril
    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
Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-19_8f1tkrj4
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 5, PDB Name: DB1_PDB3

Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-20_8k1tkrjb
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 5, PDB Name: DB1_PDB3

Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-21_7j1tkqk6
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-22_7k1tkqk6
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-23_7l1tkqk6
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-24_7m1tkqk6
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-25_7n1tkqrh
Tag: ORACOPYBKUPONFSXN_LEVEL_0
Container ID: 3, PDB Name: DB1_PDB1

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-
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-UNDOTBS1_FNO-7_8h1tkrj9"
datafile 8 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_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-SYSAUX_FNO-13_871tkrhr"
datafile 14 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-14_8c1tkril"
datafile 15 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-15_881tkrja"
datafile 16 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-16_8e1tkril"
datafile 17 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-17_881tkrhr"
datafile 19 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-UNDOTBS1_FNO-19_8f1tkrj4"
datafile 20 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-USERS_FNO-20_8k1tkrjb"
datafile 21 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-21_7j1tkqk6"
datafile 22 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-22_7k1tkqk6"
datafile 23 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-23_7l1tkqk6"
datafile 24 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-24_7m1tkqk6"
datafile 25 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-25_7n1tkqrh"
datafile 26 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-26_7o1tkqrj"
datafile 27 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-27_7p1tkqrq"
datafile 28 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-28_7q1tkqs1"
datafile 29 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-29_7r1tkr32"
datafile 30 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-30_7s1tkr3a"
datafile 31 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-31_7t1tkr3i"
datafile 32 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-32_7u1tkr42"
datafile 33 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-33_7v1tkra6"
datafile 34 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-34_801tkram"
datafile 35 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-35_811tkrap"

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_l7vzdfwm_.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_l7vxylwp_.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_l7vykf23_.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_l7vyrm5_.arc thread=1 sequence=202
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_06_05/o1_mf_1_203_l7vzdfwm_.arc thread=1 sequence=203
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*Plus: Release 19.0.0.0.0 - Production on Wed Jun 7 15:58:12 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 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_l81bhz6g_.tmp
/nfsfsxn/oracopy/DB1/FB867DA8C68C816EE053630F1EAC2BCF/datafile/o1_mf_temp_l81bj16t_.tmp
/nfsfsxn/oracopy/DB1/FB867EA89ECF81C0E053630F1EACB901/datafile/o1_mf_temp_l81bj135_.tmp
/nfsfsxn/oracopy/DB1/FB867F8A4D4F821CE053630F1EAC69CC/datafile/o1_mf_temp_l81bj13g_.tmp
/nfsfsxn/oracopy/DB1/datafile/o1_mf_temp_l81bhwjg_.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_7n1tkqhr
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-26_7o1tkqrj
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-27_7p1tkqr
/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_861tkrgo
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-14_871tkhr
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-18_881tkhr

NAME
----------------------------------

/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-20_891tkhr

NAME
----------------------------------

/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-22_8a1tkhr

NAME
----------------------------------

/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-24_8b1tkhr

NAME
----------------------------------

/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-26_8c1tkhr
NAME

--------------------------------------------------------------------

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;

    ID  DT
----------

---

43 rows selected.
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> 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
Connected to database DB1 (DBID=1730530050)
```
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-SYSAUX_FNO-5_8d1tkri - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-6_891tkrh - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-7_8h1tkrj - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-8_8gltkrj - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-9_8a1tkrh - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-10_861tkrg - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-11_8c1tkri - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-12_8b1tkrh - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-13_871tkrg - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-14_8e1tkri - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-15_8d1tkrh - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-16_8a1tkrg - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-17_8c1tkri - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-18_881tkrh - dbid changed, wrote new name
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;

NAME      OPEN_MODE            LOG_MODE
--------- -------------------- ------------
DB1TST    READ WRITE           NOARCHIVELOG

SQL> show pdbs

CON_ID CON_NAME                       OPEN_MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
  2 PDB$SEED                       READ ONLY  NO
  3 DB1_PDB1                        MOUNTED
  4 DB1_PDB2                        MOUNTED
  5 DB1_PDB3                        MOUNTED

SQL> alter pluggable database all open;
Pluggable database altered.

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

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-8225-4784bced9843&sc_channel=ps&s_kwcid=AL!4422!3!467723097970le!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfbhCKARlsAkJzSrdrqWQqghn6i71jWzSeaT9Uh1-vY-VfhJixF-nxv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:G:s&s_kwcid=AL!4422!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

[Diagram of Oracle Deployment in AWS FSx/EC2 with NFS/ASM]

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>Software</td>
<td>Version</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------</td>
<td>----------------------------------------------------------------------------</td>
<td></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>
<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>
<td></td>
</tr>
<tr>
<td>Oracle Grid Infrastructure</td>
<td>Version 19.18</td>
<td>Applied RU patch p34762026_190000_Linux-x86-64.zip</td>
<td></td>
</tr>
<tr>
<td>Oracle Database</td>
<td>Version 19.18</td>
<td>Applied RU patch p34765931_190000_Linux-x86-64.zip</td>
<td></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>
<td></td>
</tr>
<tr>
<td>SnapCenter Service</td>
<td>Version</td>
<td>v2.3.1.2324</td>
<td></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
particular importance 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.

```
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  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.
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 " ".

```bash
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

```bash
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).

```bash
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.

   ```
   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 -junction-path /ora_01_biny -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 -junction-path /ora_01_data -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 -junction-path /ora_01_logs -snapshot-policy none -tiering-policy snapshot-only
   ```

5. Validate the DB volumes created.

   ```
   vol show
   ```

   This is expected to return:
Database storage configuration

<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>47.50GB</td>
<td>0%</td>
<td></td>
<td></td>
<td></td>
<td></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>95.00GB</td>
<td>0%</td>
<td></td>
<td></td>
<td></td>
<td></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>95.00GB</td>
<td>0%</td>
<td></td>
<td></td>
<td></td>
<td></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>972.1MB</td>
<td>0%</td>
<td></td>
<td></td>
<td></td>
<td></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

```bash
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.

```bash
sudo chown oracle:oinstall /oralogs
```

11. Add a mount point to /etc/fstab.

```bash
sudo vi /etc/fstab
```

Add the following line.

```ini
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

```bash
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`.

```bash
sudo vi /etc/ssh/sshd_config
```

2. Restart the sshd service.

```bash
sudo systemctl restart sshd
```

3. Reset the Oracle user password.

```bash
sudo passwd oracle
```

4. Log in as the Oracle Restart software owner user (oracle). Create an Oracle directory as follows:

```bash
mkdir -p /u01/app/oracle
mkdir -p /u01/app/oraInventory
```

5. Change the directory permission setting.

```bash
chmod -R 775 /u01/app
```

6. Create a grid home directory and change to it.

```bash
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.

```bash
unzip -q /tmp/archive/Linux.X64_193000_grid_home.zip
```

8. From grid home, delete the OPatch directory.

```bash
rm -rf OPatch
```

9. From grid home, copy `p6880880_190000_Linux-x86_64.zip` to the grid_home, and then unzip it.
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>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>ora.LISTENER.lsnr</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>ora.asm</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
<tr>
<td></td>
<td>Started,STABLE</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>ora.diskmon</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>ora.evmd</td>
<td>ONLINE</td>
<td>ONLINE</td>
<td>ip-172-30-15-58</td>
</tr>
</tbody>
</table>

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 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.db1.log"
for further details.

13. As Oracle user, validate Oracle Restart HA services after DB creation.
```
[oracle@ip-172-30-15-58 db1]$ ../grid/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
  1 ONLINE ONLINE       ip-172-30-15-58          STABLE
ora.db1.db
  1 ONLINE ONLINE       ip-172-30-15-58
  Open,HOME=/u01/app/oracle/product/19.0.0
/orb1,STABLE
ora.diskmon
  1 OFFLINE OFFLINE     STABLE
ora.evmd
  1 ONLINE ONLINE       ip-172-30-15-58          STABLE
----------
[oracle@ip-172-30-15-58 db1]$`
15. Add following entries:

```bash
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/undotbs1.267.1132177853
+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.274.1132171298
```
NAME

+DATA/DB1/F785288B8CD1BA78E0533A0F1EACC6DF/DATAFILE/system.276.11321
+DATA/DB1/F785288B8CD1BA78E0533A0F1EACC6DF/DATAFILE/sysaux.277.11321
+DATA/DB1/F785288B8CD1BA78E0533A0F1EACC6DF/DATAFILE/undotbs1.275.113
+DATA/DB1/F785288B8CD1BA78E0533A0F1EACC6DF/DATAFILE/users.279.11321
+DATA/DB1/F785288B8CD1BA78E0533A0F1EACC6DF/DATAFILE/system.277.11321
+DATA/DB1/F785288B8CD1BA78E0533A0F1EACC6DF/DATAFILE/sysaux.277.11321
+DATA/DB1/F785288B8CD1BA78E0533A0F1EACC6DF/DATAFILE/undotbs1.275.113
+DATA/DB1/F785288B8CD1BA78E0533A0F1EACC6DF/DATAFILE/users.279.11321
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>

SQL>

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
SQL> select name, log_mode from v$database;

NAME      LOG_MODE
--------- ------------
DB1       ARCHIVELOG

SQL> select svrname, dirname from v$dnfs_servers;

SVRNAME
---------
fsx_01

DIRNAME
---------
/ora_01_data
/ora_01_biny
/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
--------- ------------

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-8225-4784bced9843&sc_channel=aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARIsAjzSrdqwQrghn6I71jjiWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_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](image)

Hardware and software components

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSx ONTAP storage</td>
<td>Current version offered by AWS</td>
</tr>
<tr>
<td>EC2 instance for compute</td>
<td>t2.xlarge/4vCPU/16G</td>
</tr>
<tr>
<td>One FSx HA cluster in the same VPC and availability zone</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>Software</th>
<th>Version/Details</th>
<th>Notes</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 v2.3.1.2324</td>
<td></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.

   ```bash
   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.

   ```bash
   [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.

   ```bash
   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/transient_hugepage/enabled
    echo never > /sys/kernel/mm/transient_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

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.timeoreplacement_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.

```bash
systemctl enable iscsid
systemctl start iscsid
```

17. Retrieve the iSCSI initiator address to be used for database LUN mapping.

```bash
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).

```
usermod -a -G asm oracle
```

20. Stop and disable Linux firewall if it is active.

```
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.

```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 -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 -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 -snapshot-policy none -tiering-policy snapshot-only
```

5. Create a binary LUN within the database binary volume.

```bash
lun create -path /vol/ora_01_biny/ora_01_biny_01 -size 40G -ostype linux
```

6. Create data LUNs within the database data volume.

```bash
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>Protocol</th>
<th>Path</th>
<th>Igroup</th>
<th>LUN ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>iscsi</td>
<td>/vol/ora_01_biny/ora_01_biny_01</td>
<td>ora_01</td>
<td>0</td>
</tr>
<tr>
<td>iscsi</td>
<td>/vol/ora_01_data/ora_01_data_01</td>
<td>ora_01</td>
<td>1</td>
</tr>
<tr>
<td>iscsi</td>
<td>/vol/ora_01_data/ora_01_data_02</td>
<td>ora_01</td>
<td>2</td>
</tr>
<tr>
<td>iscsi</td>
<td>/vol/ora_01_data/ora_01_data_03</td>
<td>ora_01</td>
<td>3</td>
</tr>
<tr>
<td>iscsi</td>
<td>/vol/ora_01_data/ora_01_data_04</td>
<td>ora_01</td>
<td>4</td>
</tr>
<tr>
<td>iscsi</td>
<td>/vol/ora_01_logs/ora_01_logs_01</td>
<td>ora_01</td>
<td>5</td>
</tr>
<tr>
<td>iscsi</td>
<td>/vol/ora_01_logs/ora_01_logs_02</td>
<td>ora_01</td>
<td>6</td>
</tr>
</tbody>
</table>

Database storage configuration

333
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 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.

```bash
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.

```
[ec2-user@ip-172-30-15-58 ~]$ sudo sanlun lun show
controller(7mode/E-Series)/  device
host       lun
vserver(cDOT/FlashRay) lun-pathname
filename      adapter    protocol   size    product
svm_ora       /vol/ora_01_logs/ora_01_logs_02
/dev/sdn      host3      iSCSI      40g     cDOT
svm_ora       /vol/ora_01_logs/ora_01_logs_01
/dev/sdm      host3      iSCSI      40g     cDOT
svm_ora       /vol/ora_01_data/ora_01_data_03
/dev/sdk      host3      iSCSI      20g     cDOT
svm_ora       /vol/ora_01_data/ora_01_data_04
/dev/sdl      host3      iSCSI      20g     cDOT
svm_ora       /vol/ora_01_data/ora_01_data_01
/dev/sdj      host3      iSCSI      20g     cDOT
svm_ora       /vol/ora_01_biny/ora_01_biny_01
/dev/SDH      host3      iSCSI      40g     cDOT
svm_ora       /vol/ora_01_logs/ora_01_logs_02
/dev/SDG      host2      iSCSI      40g     cDOT
svm_ora       /vol/ora_01_logs/ora_01_logs_01
/dev/SDF      host2      iSCSI      40g     cDOT
svm_ora       /vol/ora_01_data/ora_01_data_04
/dev/SDE      host2      iSCSI      20g     cDOT
svm_ora       /vol/ora_01_data/ora_01_data_02
/dev/SDC      host2      iSCSI      20g     cDOT
svm_ora       /vol/ora_01_data/ora_01_data_03
/dev/SDD      host2      iSCSI      20g     cDOT
svm_ora       /vol/ora_01_biny/ora_01_biny_01
/dev/SDA      host2      iSCSI      40g     cDOT
```
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
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 20:13 3600a09806c574235472455534e685146 -> ../dm-6
crw------- 1 root root 10, 236 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
and return as follow:

```bash
FsxId02ad7bf3476b741df::> lun show -fields serial-hex
vserver path serial-hex
------- ------------------------------- ------------------------
svm_ora /vol/ora_01_biny/ora_01_biny_01 6c574235472455534e68512d
svm_ora /vol/ora_01_data/ora_01_data_01 6c574235472455534e685141
svm_ora /vol/ora_01_data/ora_01_data_02 6c574235472455534e685142
svm_ora /vol/ora_01_data/ora_01_data_03 6c574235472455534e685143
svm_ora /vol/ora_01_data/ora_01_data_04 6c574235472455534e685144
svm_ora /vol/ora_01_logs/ora_01_logs_01 6c574235472455534e685145
svm_ora /vol/ora_01_logs/ora_01_logs_02 6c574235472455534e685146
7 entries were displayed.
```

9. Update the `/dev/multipath.conf` file to add a user-friendly name for the multipath device.

```bash
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.

```bash
sudo fdisk /dev/mapper/ora_01_biny_01
```

12. Format the partitioned binary LUN with an XFS file system.

```bash
sudo mkfs.xfs /dev/mapper/ora_01_biny_01p1
```

13. Mount the binary LUN to /u01.

```bash
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.

```bash
sudo chown oracle:oinstall /u01
```

15. Find the UUI of the binary LUN.

```bash
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`.
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:

```plaintext
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`.

```bash
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.

```
rpm -ivh /u01/app/oracle/product/19.0.0/grid/cv/rpm/cvuqdisk-1.0.10-1.rpm
```

15. Unset $ORACLE_BASE.

```
unset ORACLE_BASE
```

16. Log into the EC2 instance as the Oracle user and extract the patch in the /tmp/archive folder.

```
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.

```
./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):

```
/u01/app/oraInventory/orainstRoot.sh

/u01/app/oracle/product/19.0.0/grid/root.sh
```

19. As root user, reload the multipathd.
systemctl restart multipathd

20. As the Oracle user, execute the following command to complete the configuration:

```bash
/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.

```bash
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.

```bash
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@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'

Oracle database installation
1. Log in as the Oracle user and unset `$ORACLE_HOME` and `$ORACLE_SID` if it is set.

   ```
   unset ORACLE_HOME
   unset ORACLE_SID
   ```

2. Create the Oracle DB home directory and change to it.

   ```
   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.

   ```
   unzip -q /tmp/archive/LINUX.X64_193000_db_home.zip
   ```

4. From the DB home, delete the OPatch directory.

   ```
   rm -rf OPatch
   ```

5. From DB home, unzip `p6880880_190000_Linux-x86-64.zip`.

   ```
   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`.

   ```
   vi cv/admin/cvu_config
   ```

7. From the `/tmp/archive` directory, unpack the DB 19.18 RU patch.

   ```
   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:
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.
[oracle@ip-172-30-15-58 db1]$ ../grid/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.db1.db        ONLINE  ONLINE       ip-172-30-15-58       Open,HOME=/u01/app/oracle/product/19.0.0/db1,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

14. Set the Oracle user `.bash_profile`.

```bash
vi ~/.bash_profile
```

15. Add following entries:

```bash
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
/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/DATAFILE/users.259.1132176247
+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/undotbs1.270.11321777853
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/users.274.1132177871

NAME
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/system.276.1132177871
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/sysaux.277.1132177781
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/undotbs1.275.1132177781
+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.

```sql
alter system set db_recovery_file_dest_size = 80G scope=both;
```

18. Log into the database with sqlplus and enable archive log mode.

```sql
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.
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
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=6536</td>
</tr>
<tr>
<td>Data files</td>
<td>rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=6536</td>
</tr>
<tr>
<td>Redo logs</td>
<td>rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=6536</td>
</tr>
<tr>
<td>ORACLE_HOME</td>
<td>rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=6536</td>
</tr>
<tr>
<td>ORACLE_BASE</td>
<td>rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=6536</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**

Refereed 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.

<table>
<thead>
<tr>
<th>fsx (svm-005c6edf027866ca4)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary</strong></td>
</tr>
<tr>
<td>SVM ID: svm-005c6edf027866ca4</td>
</tr>
<tr>
<td>SVM name: fx</td>
</tr>
<tr>
<td>Status: Available</td>
</tr>
<tr>
<td>Life cycle state: Created</td>
</tr>
<tr>
<td>Life cycle: Created</td>
</tr>
<tr>
<td>Life cycle: CREATED</td>
</tr>
<tr>
<td>Life cycle: Created</td>
</tr>
<tr>
<td>Life cycle: Created</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Default storage virtual machine configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Storage virtual machine name</strong></td>
</tr>
<tr>
<td>f sx ora_ prod</td>
</tr>
<tr>
<td><strong>SVM administrative password</strong></td>
</tr>
<tr>
<td>Password for this SVM's &quot;vsadmin&quot; user, which you can use to access the ONTAP CLI or REST API.</td>
</tr>
<tr>
<td>○ Don't specify a password</td>
</tr>
<tr>
<td>○ Specify a password</td>
</tr>
<tr>
<td>Password</td>
</tr>
<tr>
<td>Confirm password</td>
</tr>
<tr>
<td>Active Directory</td>
</tr>
<tr>
<td>Joining an Active Directory enables access from Windows and MacOS clients over the SMB protocol.</td>
</tr>
<tr>
<td>○ Do not join an Active Directory</td>
</tr>
<tr>
<td>○ Join an Active Directory</td>
</tr>
</tbody>
</table>

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-005c6edf027865ca4 | 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-0a51a3f08922222d5 | 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

Cancel Confirm
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.

```
git clone https://github.com/NetApp-Automation/na_oracle19c_deploy.git
```

```
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.

```
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.

```
ansible-playbook -i hosts rds_preclone_config.yml -u ec2-user --private -key accesststkey.pem -e @vars/fsx_vars.yml -t requirements_config
```

```
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.

```
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.

```
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.

```
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`, and `standby_cluster_name` with the appropriate names for your environment.

```
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.

```bash
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.

```bash
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.

---

Image: Credential dialog box with default settings.
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 that 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.

   ```bash
   [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.

<table>
<thead>
<tr>
<th>Summary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup name</td>
<td>ip-10-0-0-151_03-25-2022_11.15.01.1503_0</td>
</tr>
<tr>
<td>Backup date</td>
<td>03/25/2022 11:15:11 AM</td>
</tr>
<tr>
<td>Restore scope</td>
<td>All DataFiles</td>
</tr>
<tr>
<td>Recovery scope</td>
<td>Until SCN 1778546</td>
</tr>
<tr>
<td>Auxiliary</td>
<td>Change database state if necessary, Open the database or container database in READ-WRITE mode after recovery</td>
</tr>
<tr>
<td>destination</td>
<td></td>
</tr>
<tr>
<td>Prescript full path</td>
<td>None</td>
</tr>
<tr>
<td>Prescript arguments</td>
<td></td>
</tr>
<tr>
<td>Postscript full path</td>
<td>None</td>
</tr>
<tr>
<td>Postscript arguments</td>
<td></td>
</tr>
<tr>
<td>Send email</td>
<td>No</td>
</tr>
</tbody>
</table>
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:

```
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.

```bash
$ 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.

*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 an 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 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, 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 10 x 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
- The quota assigned to the volume.

The following diagram shows how the throughput limit for an Oracle database volume is calculated.

![Diagram showing throughput calculation](image)

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.

![Diagram demonstrating SLOB benchmark](image)

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.

  ```bash
  sysctl -a | grep tcp.*.slot_table
  ```

- The following table provides recommended NFS mount options for a single instance of Linux NFSv3.

```bash
<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protos</td>
<td>tcp,rmap,udata,ufcp,rsa,udp,udp6,udraw,uscsi,sv4,sv41,sv42,rsync,iso,iso9660</td>
</tr>
<tr>
<td>Server Equity</td>
<td>10</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.

### 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**.
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

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription

Resource group

Instance details

Virtual machine name

Region

Availability options

Security type

Image

Run with Azure Spot discount

Size

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.
Create a virtual machine

Basics  Disks  Networking  Management  Advanced  Tags  Review + create

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. Learn more of

Disk options

OS disk type *  ○

Delete with VM  ○

Enable encryption at host  ○

Encryption at host is not registered for the selected subscription. Learn more about enabling this feature

Encryption type *

(Default) Encryption at-rest with a platform-managed key

Enable Ultra Disk compatibility  ○

Data disks for acao-ora01

You can add and configure additional data disks for your virtual machine or attach existing disks. This VM also comes with a temporary disk.

<table>
<thead>
<tr>
<th>LUN</th>
<th>Name</th>
<th>Size (GiB)</th>
<th>Disk type</th>
<th>Host caching</th>
<th>Delete with VM</th>
</tr>
</thead>
</table>

Create and attach a new disk  Attach an existing disk

Advanced

Review + create  < Previous  Next : Networking >

4. Choose the VNet and subnet. Allocate a public IP for external VM access. Then go to the Management page.
Create a virtual machine

Network interface

When creating a virtual machine, a network interface will be created for you.

- **Virtual network**: ANFAVStVal
  - *Create new*

- **Subnet**: VM_Sub (172.30.137.128/25)
  - *Manage subnet configuration*

- **Public IP**: (new) acco-ora01-ip
  - *Create new*

- **NIC network security group**: None
  - **Basic**
  - **Advanced**

- **Public inbound ports**: None
  - **Allow selected ports**

- **Select inbound ports**: SSH (22)
  - *This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.*

- **Delete public IP and NIC when VM is deleted**: Yes

- **Enable accelerated networking**: Yes

**Load balancing**

You can place this virtual machine in the backend pool of an existing Azure load balancing solution. [Learn more](#)

- **Place this virtual machine behind an existing load balancing solution**: No

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.

```bash
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.

```
./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.

```
su – oracle
sqlplus / AS SYSDBA
```

2. Create the user.

```
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.

    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.

    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.
5. Create the `tnsnames.ora` file. In the following example command, HOST should be set to the IP address of the Oracle Database and the Server SID should be set to the Oracle Database SID.

```bash
echo "# Connection string
AZACSNAP="(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=172.30.137.142)(PORT=1521))(CONNECT_DATA=(SID=ORATST)))"
" > $TNS_ADMIN/tnsnames.ora
```

6. Create the `sqlnet.ora` file.

```bash
echo "SQLNET.WALLET_OVERRIDE = TRUE
WALLET_LOCATION=(
    SOURCE=(METHOD=FILE)
    (METHOD_DATA=(DIRECTORY="$TNS_ADMIN/.oracle_wallet"))
) " > $TNS_ADMIN/sqlnet.ora
```

7. Test Oracle access using the wallet.

```bash
sqlplus /@AZACSNAP as SYSBACKUP
```

The expected output from the command:

```bash
[azacsnap@acao-ora01 ~]$ sqlplus /@AZACSNAP as SYSBACKUP

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Sep 8 18:02:07 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>
```

**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/netAppAc
   counts/.../capacityPools/Premium/volumes/...): /subscriptions/0efa2dfb-917c-4497-b56a-
   b3f4eadb8111/resourceGroups/ANFAVSRG/providers/Microsoft.NetApp/netAppAc
   counts/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/netAppA
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:
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:

```
azacsnap -c backup --volume data --prefix ora_test --retention=1
```
# /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@acao-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(
    id integer,
    event varchar(100),
    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;

<table>
<thead>
<tr>
<th>ID</th>
<th>EVENT</th>
<th>DT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>insert a data marker to validate snapshot restore</td>
<td>12-SEP-22 07.07.35.000000 PM</td>
</tr>
</tbody>
</table>

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__29h6qqyw_.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_43__22rnjq9q_.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.

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.

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.

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.
Getting started

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>
<tr>
<td>Role</td>
<td>Tasks</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<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:
   
   ```
   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:
   
   ```
   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.

   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

```yaml
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.

```yaml
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.

```yaml
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.

```yaml
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.

```
#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: "eOM"}

# 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.
```sql
[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.
[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.

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:

```yaml
#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

```yaml
######################################################################
##############      Host Variables Configuration        ##############
######################################################################
# 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"
ple_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
# Oracle 19c deployment global user configuration variables
# Consolidate all variables from ontap, linux and oracle

# 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 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"}

#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:
### DB env specific install and config variables ###

```
# 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"
```

---

### 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.

<i> <username> must be changed to match your environment. 

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;

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.
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>ontap_setup</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>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</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</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>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 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.
# Oracle Data Protection global user configuration variables
# Ontap env specific config variables
hosts_group: "ontap"
ca_signed_certs: "false"

# Inter-cluster LIF details
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

# Inventory group name
# Default inventory group name - "ontap"
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: "123123123123123123123123123123123"

# 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"
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"
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.

   ![Info](https://via.placeholder.com/15)
   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.
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.

TR-4794: Oracle databases on NetApp EF-Series
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.
Add a SQL Server host to 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 in 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.

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:


- 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.

3. Specify the password for fsxadmin.

4. Specify the password for SVMs.

5. 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.

6. Use the following SSH command to create volumes:

```bash
vol create -vserver svm001 -volume vol_awsslprod01_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_awsslprod01_data -fractional-reserve 0
volume modify -vserver svm001 -volume vol_awsslprod01_data -space -mgmt-try-first vol_grow
volume snapshot autodelete modify -vserver svm001 -volume vol_awsslprod01_data -delete-order oldest_first
```

7. Start the iSCSI service with PowerShell using elevated privileges in Windows Servers.

```bash
Start-service -Name msiscsi
Set-Service -Name msiscsi -StartupType Automatic
```


```bash
Install-WindowsFeature -name Multipath-IO -Restart
```


```bash
Get-InitiatorPort | select NodeAddress
```
10. Connect to Storage virtual machines (SVM) using putty and create an iGroup.

```bash
```

11. 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
```

12. To achieve I/O alignment with the OS partitioning scheme, use windows_2008 as the recommended LUN type. Refer here for additional information.

13. Use the following SSH command to map igroup to the LUNs that you just created.

```bash
lun show
lun map -path /vol/vol_awssqlprod01_data/lun_awssqlprod01_data -igroup igrp_awssqlprod01
lun map -path /vol/vol_awssqlprod01_log/lun_awssqlprod01_log -igroup igrp_awssqlprod01
```
14. 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.

15. Connect Windows Server to an SVM with an iSCSI target. Find the target IP address from AWS Portal.

```
svmsql (svm-09e98ab33a31b724a)
```

16. 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 select OK to close all windows.
17. Repeat step 12 for the second iSCSI IP from the SVM.

18. Select the **Targets** tab, select **Connect**, and select **Enable muti-path**.

19. 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}

InitiatorInstanceName          : 
InitiatorPortalAddress         : 
ISDataDigest                  : False
ISHeaderDigest                : False
TargetPortalAddress           : 10.2.1.167
TargetPortalPortNumber        : 3260
PSCoordinateName              : 

InitiatorInstanceName          : 
InitiatorPortalAddress         : 
ISDataDigest                  : False
ISHeaderDigest                : False
TargetPortalAddress           : 10.2.2.12
TargetPortalPortNumber        : 3260
PSCoordinateName              : 

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.

$targets = Get-IscsiTarget
foreach ($target in $targets)
{
    Connect-IscsiTarget -IsMultipathEnabled $true -NodeAddress $target.NodeAddress -IsPersistent $true
}

2. Initialize disks with the following PowerShell command.
```powershell
$disks = Get-Disk | where PartitionStyle -eq raw
foreach ($disk in $disks) {Initialize-Disk $disk.Number}
```

3. Run the Create Partition and Format Disk commands with PowerShell.

```
New-Partition -DiskNumber 1 -DriveLetter F -UseMaximumSize
Format-Volume -DriveLetter F -FileSystem NTFS -AllocationUnitSize 65536
New-Partition -DiskNumber 2 -DriveLetter G -UseMaximumSize
Format-Volume -DriveLetter G -FileSystem NTFS -AllocationUnitSize 65536
```

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>
</tbody>
</table>
| Memory                      | Minimum: 8GB  
                                  Recommended: 32GB                                         |
| Storage space               | Minimum space for installation: 10GB  
                                  Minimum space for repository: 10GB                           |
| Supported operating system  | • Windows Server 2012  
                                  • Windows Server 2012 R2   
                                  • Windows Server 2016  
                                  • Windows Server 2019                                           |
| Software packages           | • .NET 4.5.2 or later  
                                  • Windows Management Framework (WMF) 4.0 or later  
                                  • PowerShell 4.0 or later                                       |

For detailed information, refer to space and sizing requirements.

For version compatibility, see the NetApp Interoperability Matrix Tool.
Database storage layout

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.

2. Add the storage system by selecting Storage Systems > New and provide the appropriate FSx for ONTAP storage information.

3. 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:

\[
\text{Size of host log directory} = \text{system database size} + (\text{maximum DB LDF size} \times \text{daily log change rate \%} \times \text{(Snapshot copy retention)} + (1 – \text{LUN overhead space \%})
\]

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.
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 step 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 step 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.
7. Follow the Clone from Backup instructions by providing the new SQL Server instance on EC2 and instance name to finish the clone process.
8. 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

```powershell
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

```powershell
....
#Enable MPIO and Start iSCSI Service
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.datavolsize;lunname=('l_'+$hostname+'_data'+[string]$i);lunsize=$fsxN.datalunsize})
  }
}
$secStringPassword = ConvertTo-SecureString $fsxN.password
-AsPlainText -Force
($fsxN.login, $secStringPassword)
$igroup = 'igrp_'+$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" 
    $sshcmd += 
    "#-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'; logv
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.

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 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; 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:

```plaintext
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=ZxRUJwFlLXgHS8As7HzXJOaDXXVJ7PxxIX3ACpx56XY%3D"
echo %dest%
```

The following example cmd is executed in PowerShell:

```
-recursive
```
INFO: Scanning...
INFO: Any empty folders will not be processed, because source and/or destination doesn't have full folder support
Job b3731dd8-da61-9441-7281-17a4db09ce30 has started
Log file is located at: C:\Users\niyaz\.azcopy\b3731dd8-da61-9441-7281-17a4db09ce30.log
0.0 %, 0 Done, 0 Failed, 2 Pending, 0 Skipped, 2 Total,
INFO: azcopy.exe: A newer version 10.10.0 is available to download
0.0 %, 0 Done, 0 Failed, 2 Pending, 0 Skipped, 2 Total,
Job b3731dd8-da61-9441-7281-17a4db09ce30 summary
Elapsed Time (Minutes): 0.0333
Number of File Transfers: 2
Number of Folder Property Transfers: 0
Total Number of Transfers: 2
Number of Transfers Completed: 2
Number of Transfers Failed: 0
Number of Transfers Skipped: 0
TotalBytesTransferred: 5
Final Job Status: Completed

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.

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><strong>FSx ONTAP storage</strong></td>
<td><strong>Current version</strong></td>
</tr>
<tr>
<td>Two FSx HA pairs in the same VPC and availability zone as primary and standby HA clusters</td>
<td></td>
</tr>
<tr>
<td><strong>EC2 instance for compute</strong></td>
<td><strong>t2.xlarge/4vCPU/16G</strong></td>
</tr>
<tr>
<td>Two EC2 T2 xlarge as primary and standby compute instances</td>
<td></td>
</tr>
<tr>
<td><strong>Ansible controller</strong></td>
<td><strong>on-prem Centos VM/4vCPU/8G</strong></td>
</tr>
<tr>
<td>A VM to host Ansible automation controller either on-premise or in the cloud</td>
<td></td>
</tr>
</tbody>
</table>

590
RedHat Linux  | RHEL-8.6.0_HVM-20220503-x86_64-2-Hourly2-GP2 | Deployed RedHat subscription for testing
---|---|---
Centos Linux  | CentOS Linux release 8.2.2004 (Core) | Hosting Ansible controller deployed in on-premises lab
PostgreSQL  | Version 14.5 | Automation pulls the latest available version of PostgreSQL from the postgresql.ora yum repo
Ansible  | Version 2.10.3 | Prerequisites for required collections and libraries installed with requirements playbook

**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 alway 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
database backup that can be ported to S3 blob storage.

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 ###
# Variables for SnapMirror Peering

# Passphrase for cluster peering authentication
passphrase: "xxxxxxx"

# Please enter destination or standby FSx cluster name
dst_cluster_name: "FsxId0cf8e0bccb14805e8"

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"

# Please enter source or primary FSx cluster name
src_cluster_name: "FsxId0cf8e0bccb14805e8"

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"
### Linux env specific config variables ###

#### NFS Mount points for PostgreSQL DB volumes

```bash
mount_points:
  - "/pgdata"
  - "/pglogs"
```

# RedHat subscription username and password

```bash
redhat_sub_username: "xxxxx"
redhat_sub_password: "xxxxx"
```

### DB env specific install and config variables ###

# The latest version of PostgreSQL RPM is pulled-installed and config file is deployed from a preconfigured template
# Recovery type and point: default as all logs and promote and leave all PITR parameters blank

**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.
ansible-playbook -i hosts postgresql_deploy.yml -u ec2-user --private-key psql_01p.pem -e @vars/fsx_vars.yml

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.

```bash
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.

```bash
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
  https://aws.amazon.com/fsx/netapp-ontap/

- Amazon EC2
  https://aws.amazon.com/pm/ec2/?trk=36c6da98-7b20-48fa-8225-4784bced9843&sc_channel=ps&s_kwcid=ALI4422i3!467723097970le!!gl!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARisAJzSrQwQrQh6i71jiWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:s&s_kwcid=ALI4422i3!467723097970le!!gl!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

SnapCenter Oracle Clone Lifecycle Automation

Allen Cao, Niyaz Mohamed, NetApp

Purpose

Customers love the FlexClone feature of NetApp ONTAP storage for databases with significant storage cost savings. This Ansible based toolkit automates the setup, cloning, and refreshing of cloned Oracle databases on schedule using the NetApp SnapCenter command line utilities for streamlined lifecycle management. The toolkit is applicable to Oracle databases deployed to ONTAP storage either on-premises or public cloud and managed by NetApp SnapCenter UI tool.

This solution addresses the following use cases:

- Setup Oracle database clone-specification configuration file.
- Create and refresh clone Oracle database on user defined schedule.

Audience

This solution is intended for the following people:

- A DBA who manages Oracle databases with SnapCenter.
- A storage administrator who manages ONTAP storage with SnapCenter.
- An application owner who has access to SnapCenter UI.

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 controller:**
- Ansible v.2.10 and higher
- ONTAP collection 21.19.1
- Python 3
- Python libraries:
  - netapp-lib
  - xmltodict
  - jmespath

**SnapCenter server:**
- version 5.0
- backup policy configured
- Source database protected with a backup policy

**Oracle servers:**
- Source server managed by SnapCenter
- Target server managed by SnapCenter
- Target server with identical Oracle software stack as source server installed and configured

**Download the toolkit**

```
git clone https://bitbucket.ngage.netapp.com/scm/ns-bb/na_oracle_clone_lifecycle.git
```

**Ansible target hosts file configuration**
The toolkit includes a hosts file which defines the targets that an Ansible playbook running against. Usually, it is the target Oracle clone hosts. Following is an example file. A host entry includes target host IP address as well as ssh key for an admin user access to the host to execute clone or refresh command.

# Oracle clone hosts

```
[clone_1]
ora_04.cie.netapp.com ansible_host=10.61.180.29
ansible_ssh_private_key_file=ora_04.pem
```

```
[clone_2]
```

```
[clone_3]
```

Global variables configuration

The Ansible playbooks take variable inputs from several variable files. Below is an example global variable file vars.yml.

```
# ONTAP specific config variables

# SnapCtr specific config variables

snapctr_usr: xxxxxxxx
snapctr_pwd: 'xxxxxxxx'

backup_policy: 'Oracle Full offline Backup'

# Linux specific config variables

# Oracle specific config variables
```
Host variables configuration

Host variables are defined in host_vars directory named as {{ host_name }}.yml. Below is an example of target Oracle host variable file ora_04.cie.netapp.com.yml that shows typical configuration.

```yaml
# User configurable Oracle clone db host specific parameters

# Source database to clone from
source_db_sid: NTAP1
source_db_host: ora_03.cie.netapp.com

# Clone database
clone_db_sid: NTAP1DEV

snapctr_obj_id: '{{ source_db_host }}{{ source_db_sid }}'
```

Additional clone target Oracle server configuration

Clone target Oracle server should have the same Oracle software stack as source Oracle server installed and patched. Oracle user .bash_profile has $ORACLE_BASE, and $ORACLE_HOME configured. Also, $ORACLE_HOME variable should match with source Oracle server setting. Following is an example.

```bash
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
  . ~/.bashrc
fi

# User specific environment and startup programs
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/NTAP1
```

Playbook execution
There are total of three playbooks to execute Oracle database clone lifecycle with SnapCenter CLI utilities.

1. Install Ansible controller prerequisites - one time only.

   ```
   ansible-playbook -i hosts ansible_requirements.yml
   ```

2. Setup clone specification file - one time only.

   ```
   ansible-playbook -i hosts clone_1_setup.yml -u admin -e @vars/vars.yml
   ```

3. Create and refresh clone database regularly from crontab with a shell script to call a refresh playbook.

   ```
   0 */4 * * * /home/admin/na_oracle_clone_lifecycle/clone_1_refresh.sh
   ```

For an additional clone database, create a separate clone_n_setup.yml and clone_n_refresh.yml, and clone_n_refresh.sh. Configure the Ansible target hosts and hostname.yml file in host_vars directory accordingly.

---

**Where to find additional information**

To learn more about the NetApp solution automation, review the following website [NetApp Solution Automation](https://www.netapp.com/index.html)  

**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**

![Architecture Diagram]

**Hardware and software components**

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azure NetApp Files</td>
<td>Current offering in Azure by Microsoft A 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</td>
</tr>
<tr>
<td>Azure VM for SnapCenter</td>
<td>Standard_B4ms - 4 vCPUs, 16GiB One Windows virtual machine instance</td>
</tr>
</tbody>
</table>
RedHat Linux | RHEL Linux 8.6 (LVM) - x64 Gen2 | Deployed RedHat subscription for testing
---|---|---
Windows Server | 2022 DataCenter; AE Hotpatch - x64 Gen2 | Hosting SnapCenter server
Oracle Database | Version 19.18 | Patch p34765931_190000_Linux-x86-64.zip
Oracle OPatch | Version 12.2.0.1.36 | Patch p6880880_190000_Linux-x86-64.zip
SnapCenter Server | Version 5.0 | Workgroup deployment
Open JDK | Version java-11-openjdk | SnapCenter plugin requirement on DB VMs
NFS | Version 3.0 | Oracle dNFS enabled
Ansible | core 2.16.2 | Python 3.6.8

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.

![Azure NetApp Files Configuration](image)

**Credential**

- **Credential Name**: `azure_anf`
- **Authentication Mode**: `Azure Credential`
- **Azure Details**
  - **Tenant ID**: Enter Tenant Id
  - **Client ID**: Enter Client Id
  - **Client Secret Key**: Enter client secret key

---

**NetApp SnapCenter**

**ADD Azure NetApp Account**

- **Credential**: `AzureAnf`
- **Subscription**: `Hybrid Cloud Tenant Orgname`
- **NetApp Account**: `ANFNS5acct (ResourceGroup: ANFNSRG)`
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 click on ‘+’ 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 blank 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.

2. Select Restore Scope. For a container database, SnapCenter is flexible to perform a full container database (All Datafiles), pluggable databases, or tablespaces 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

If you want to send notifications for Restore jobs, an SMTP server must be configured. Continue to the Summary page to save your information, and then go to Settings>Global Settings>Notification Server Settings to configure the SMTP server.
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.
```bash
[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
<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>devtmpfs</td>
<td>7.7G</td>
<td>0</td>
<td>7.7G</td>
<td>0%</td>
<td>/dev</td>
</tr>
<tr>
<td>tmpfs</td>
<td>7.8G</td>
<td>0</td>
<td>7.8G</td>
<td>0%</td>
<td>/dev/shm</td>
</tr>
<tr>
<td>tmpfs</td>
<td>7.8G</td>
<td>49M</td>
<td>7.7G</td>
<td>1%</td>
<td>/run</td>
</tr>
<tr>
<td>tmpfs</td>
<td>7.8G</td>
<td>0</td>
<td>7.8G</td>
<td>0%</td>
<td>/sys/fs/cgroup</td>
</tr>
<tr>
<td>/dev/mapper/rootvg-rootlv</td>
<td>22G</td>
<td>17G</td>
<td>5.6G</td>
<td>75%</td>
<td>/</td>
</tr>
<tr>
<td>/dev/mapper/rootvg-usrlv</td>
<td>10G</td>
<td>2.0G</td>
<td>8.1G</td>
<td>20%</td>
<td>/usr</td>
</tr>
<tr>
<td>/dev/mapper/rootvg-homelv</td>
<td>1014M</td>
<td>40M</td>
<td>975M</td>
<td>4%</td>
<td>/home</td>
</tr>
<tr>
<td>/dev/sda1</td>
<td>496M</td>
<td>106M</td>
<td>390M</td>
<td>22%</td>
<td>/boot</td>
</tr>
<tr>
<td>/dev/mapper/rootvg-varlv</td>
<td>8.0G</td>
<td>958M</td>
<td>7.1G</td>
<td>12%</td>
<td>/var</td>
</tr>
<tr>
<td>/dev/sda15</td>
<td>495M</td>
<td>5.9M</td>
<td>489M</td>
<td>2%</td>
<td>/boot/efi</td>
</tr>
<tr>
<td>/dev/mapper/rootvg-tmplv</td>
<td>12G</td>
<td>8.4G</td>
<td>3.7G</td>
<td>70%</td>
<td>/tmp</td>
</tr>
<tr>
<td>tmpfs</td>
<td>1.6G</td>
<td>0</td>
<td>1.6G</td>
<td>0%</td>
<td>/run/user/54321</td>
</tr>
<tr>
<td>172.30.136.68:/ora-02-u03</td>
<td>250G</td>
<td>2.1G</td>
<td>248G</td>
<td>1%</td>
<td>/u03</td>
</tr>
<tr>
<td>172.30.136.68:/ora-02-u01</td>
<td>100G</td>
<td>10G</td>
<td>91G</td>
<td>10%</td>
<td>/u01</td>
</tr>
<tr>
<td>172.30.136.68:/ora-02-u02</td>
<td>250G</td>
<td>7.5G</td>
<td>243G</td>
<td>3%</td>
<td>/u02</td>
</tr>
<tr>
<td>tmpfs</td>
<td>1.6G</td>
<td>0</td>
<td>1.6G</td>
<td>0%</td>
<td>/run/user/1000</td>
</tr>
<tr>
<td>tmpfs</td>
<td>1.6G</td>
<td>0</td>
<td>1.6G</td>
<td>0%</td>
<td>/run/user/0</td>
</tr>
<tr>
<td>172.30.136.68:/ora-01-u02-Clone-020624161543077</td>
<td>250G</td>
<td>8.2G</td>
<td>242G</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
4% /u02_ntap1dev

[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
SQL> select name, open_mode, log_mode from v$database;

NAME      OPEN_MODE            LOG_MODE
--------- -------------------- ------------
NTAP1DEV  READ WRITE           ARCHIVELOG

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;

NAME      OPEN_MODE            LOG_MODE
--------- -------------------- ------------
NTAP1DEV  READ WRITE           NOARCHIVELOG

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>Component</th>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azure NetApp Files storage</td>
<td>Premium Service level</td>
<td>Auto QoS type, and 4TB in storage capacity in testing</td>
</tr>
<tr>
<td>Azure instance for compute</td>
<td>Standard B4ms (4 vcpus, 16 GiB memory)</td>
<td>Two instances deployed, one as primary DB server and the other as clone DB server</td>
</tr>
</tbody>
</table>

**Software**

<table>
<thead>
<tr>
<th>Component</th>
<th>Version/Details</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>RedHat Linux</td>
<td>Red Hat Enterprise Linux 8.7 (LVM) - x64 Gen2</td>
<td>Deployed RedHat subscription for testing</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 v2.5.0-2822</td>
<td>Agent Version v2.5.0-2822</td>
</tr>
</tbody>
</table>
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.

![Credentials page](image)

2. Choose credential location as - Microsoft Azure - BlueXP.

![Add Credentials](image)

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
/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  546M  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/sda1           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  17G  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.
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.

**Oracle database clone**
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>Specification</th>
<th>Description</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 clone DB server</td>
</tr>
</tbody>
</table>

**Software**

<table>
<thead>
<tr>
<th>Component</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 v2.3.1.2324</td>
<td></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. 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**.

**AWS Authentication**

- **Region**: us-east-1 | US East (N. Virginia)

- **Select the Authentication Method**: 
  - Assume Role
  - AWS Keys

- **AWS Access Key**: [Input Field]
- **AWS Secret Key**: [Input Field]
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 **Polices** 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.

1. Select the **Basic** option if you don’t need to change any cloned database parameters.

1. 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.
Multiple entries with the same ORACLE_SID are not allowed.

```bash
SQL> select name, open_mode from v$database;
NAME    OPEN MODE
-------- ------------
DBCLONE  READ WRITE
```

SQL Plus: Release 19.0.0.0.0 - Production on Fri Mar 24 18:32:21 2023
Version 19.10.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.10.0.0.0

SQL> ```
**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
  

- Amazon FSx for NetApp ONTAP
  
  https://aws.amazon.com/fsx/netapp-ontap/

- Amazon EC2
  
  https://aws.amazon.com/pm/ec2/?trk=36c6da98-7b20-48fa-8225-4784bced9843&sc_channel=ps&s_kwcid=AL!4422!3!467723097970le!!g!aws%20ec2

**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: ./databases/TL_AWS_004 HCoD: AWS - NW,SnapCenter(OnPrem).

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
• Cloud provider permissions
• Networking for individual services

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.

The following figure shows the NetApp recommended storage layout for small or medium SQL Server database on LUNs.

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 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.

![Diagram showing Flexvol 1, 2, 3, 4 with Datafiles, Control files, REDO Logs, Archive Logs]

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.

![Diagram showing Volume Group / Disk Group with DB1:Datafiles, DB2:Datafiles, DB3:Datafiles, DB1:Logs + control file, DB2:Logs + control file, DB3:Logs + control file, Flexvol / Lun, LVM / ASM Layout]

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 sshd.conf and restarting the sshd 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.

![Confirm Fingerprint](image)

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.

![Plugin Installation Status](image)

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 screenshot]

Target CVO cluster:

![Target CVO cluster screenshot]

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.

<table>
<thead>
<tr>
<th>Retention settings</th>
<th>Daily retention settings</th>
<th>Data backup retention settings</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Snapshot copies to keep</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Keep Snapshot copies for</td>
<td>14 days</td>
</tr>
<tr>
<td>Archive Log backup retention settings</td>
<td>Total Snapshot copies to keep</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Keep Snapshot copies for</td>
<td>14 days</td>
</tr>
</tbody>
</table>
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.

---

**New SQL Server Backup Policy**

<table>
<thead>
<tr>
<th>1 Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Backup Type</td>
</tr>
<tr>
<td>3 Retention</td>
</tr>
<tr>
<td>4 Replication</td>
</tr>
<tr>
<td>5 Script</td>
</tr>
<tr>
<td>6 Verification</td>
</tr>
<tr>
<td>7 Summary</td>
</tr>
</tbody>
</table>

**Select the options to run backup verification**

Run verifications for the following backup schedules

Select how often you want the schedules to occur in the policy. The specific verification times are set at backup job creation enabling you to stagger your verification start times.

- Daily

**Database consistency checks options**

- Limit the integrity structure to physical structure of the database (PHYSICAL_ONLY)
- Suppress all information message (NO_INFOMSGS)
- Display all reported error messages per object (ALL_ERRORMSGS)
- Do not check non-clustered indexes (NOINDXR)
- Limit the checks and obtain the locks instead of using an internal database Snapshot copy (TABLOCK)

**Log backup**

- Verify log backup.

**Verification script settings**

<table>
<thead>
<tr>
<th>Script timeout</th>
<th>60 secs</th>
</tr>
</thead>
</table>

---

751
Create a database log backup policy for SQL.

1. Log into SnapCenter with a database management user ID, click Settings > Polices, and then New to launch a new policy creation workflow.
2. Define the log backup option and schedule frequency. For SQL Server configured with a 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.
6. **Summary.**
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.
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.
8. Summary.

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.
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
public cloud.

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 Lets 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

<table>
<thead>
<tr>
<th>Replication</th>
<th>On-Premises ONTAP</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>hybridcvo</td>
<td>Cloud Volumes ONTAP</td>
<td>Capacity</td>
</tr>
</tbody>
</table>

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>Mirror</td>
<td>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
reality. SnapCenter can not only protect your production database on-premises, but can also it 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: **ora_standby.demo.netapp.com**

Mount path: `/var/opt/snapcenter/sco/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><code>svm_on/plex/rhel2_u03</code></td>
<td><code>svm_hybridcvo/rhel2_u03_dr</code></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.

⚠️ If you want to send notifications for Clone jobs, an SMTP server must be configured. Continue to the Summary page to save your information, and then go to Settings>Global Settings>Notification Server Settings to configure the SMTP server.
10. 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 inquiries.

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.

```sql
create table dr_test;
```

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.
Mount backups

Choose the host to mount the backup: ora-standby.demo.netapp.com

Mount path: /var/opt/snapcenter/scv/backup_mount/rhel2_cdb2_log_09-17-2021_18.20.04.1177_1/cdb2

Secondary storage location: Snap Vault / Snap Mirror

Source Volume: svm_onPrem:rhel2_u03

Destination Volume: svm_hybridcvorrhel2_u03_dr

Mount

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.

![SnapCenter Resource Page]

3. Manually run a log backup to flush the last transaction to be replicated to secondary storage in the public cloud.

![Backup Configuration]

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 inquiries.

DB Automation Toolkits

SnapCenter Oracle Clone Lifecycle Automation

Allen Cao, Niyaz Mohamed, NetApp

Purpose

Customers love the FlexClone feature of NetApp ONTAP storage for databases with significant storage cost savings. This Ansible based toolkit automates the setup, cloning, and refreshing of cloned Oracle databases on schedule using the NetApp SnapCenter command line utilities for streamlined lifecycle management. The toolkit is applicable to Oracle databases deployed to ONTAP storage either on-premises or public cloud and managed by NetApp SnapCenter UI tool.

This solution addresses the following use cases:

• Setup Oracle database clone-specification configuration file.
• Create and refresh clone Oracle database on user defined schedule.

Audience

This solution is intended for the following people:

• A DBA who manages Oracle databases with SnapCenter.
• A storage administrator who manages ONTAP storage with SnapCenter.
• An application owner who has access to SnapCenter UI.
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 controller:**
- Ansible v.2.10 and higher
- ONTAP collection 21.19.1
- Python 3
- Python libraries:
  - netapp-lib
  - xmltodict
  - jmespath

**SnapCenter server:**
- version 5.0
- backup policy configured
- Source database protected with a backup policy

**Oracle servers:**
- Source server managed by SnapCenter
- Target server managed by SnapCenter
- Target server with identical Oracle software stack as source server installed and configured

Download the toolkit

```
git clone https://bitbucket.ngage.netapp.com/scm/ns-bb/na_oracle_clone_lifecycle.git
```
Ansible target hosts file configuration

The toolkit includes a hosts file which define the targets that an Ansible playbook running against. Usually, it is the target Oracle clone hosts. Following is an example file. A host entry includes target host IP address as well as ssh key for an admin user access to the host to execute clone or refresh command.

#Oracle clone hosts

[clone_1]
ora_04.cie.netapp.com ansible_host=10.61.180.29
ansible_ssh_private_key_file=ora_04.pem

[clone_2]

[clone_3]

Global variables configuration

The Ansible playbooks take variable inputs from several variable files. Below is an example global variable file vars.yml.

# ONTAP specific config variables

# SnapCtr specific config variables

snapctr_usr: xxxxxxxx
snapctr_pwd: 'xxxxxxxx'

backup_policy: 'Oracle Full offline Backup'

# Linux specific config variables

# Oracle specific config variables
Host variables configuration

Host variables are defined in host_vars directory named as {{ host_name }}.yml. Below is an example of target Oracle host variable file ora_04.cie.netapp.com.yml that shows typical configuration.

```yaml
# User configurable Oracle clone db host specific parameters

# Source database to clone from
source_db_sid: NTAP1
source_db_host: ora_03.cie.netapp.com

# Clone database
clone_db_sid: NTAP1DEV

snapctr_obj_id: '{{ source_db_host }}{{ source_db_sid }}'
```

Additional clone target Oracle server configuration

Clone target Oracle server should have the same Oracle software stack as source Oracle server installed and patched. Oracle user .bash_profile has $ORACLE_BASE, and $ORACLE_HOME configured. Also, $ORACLE_HOME variable should match with source Oracle server setting. Following is an example.

```bash
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/NTAP1
```

Playbook execution
There are total of three playbooks to execute Oracle database clone lifecycle with SnapCenter CLI utilities.

1. Install Ansible controller prerequisites - one time only.

   ansible-playbook -i hosts ansible_requirements.yml

2. Setup clone specification file - one time only.

   ansible-playbook -i hosts clone_1_setup.yml -u admin -e @vars/vars.yml

3. Create and refresh clone database regularly from crontab with a shell script to call a refresh playbook.

   0 */4 * * * /home/admin/na_oracle_clone_lifecycle/clone_1_refresh.sh

For an additional clone database, create a separate clone_n_setup.yml and clone_n_refresh.yml, and clone_n_refresh.sh. Configure the Ansible target hosts and hostname.yml file in host_vars directory accordingly.

Where to find additional information

To learn more about the NetApp solution automation, review the following website NetApp Solution Automation

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.

<table>
<thead>
<tr>
<th>Ansible v.2.10 and higher</th>
</tr>
</thead>
<tbody>
<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>

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)

Download the toolkit

```
git clone https://github.com/NetApp/na_ora_aws_migration.git
```

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.

```bash
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.

```bash
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.

```bash
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](https://www.netapp.com/

### 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 (HA/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.

- Ansible v.2.10 and higher
- ONTAP collection 21.19.1
- Python 3
- Python libraries:
  - netapp-lib
  - xmltodict
  - jmespath
- 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
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:

ONTAP - retrieve FSx storage parameters using AWS FSx console for both source and target FSx clusters.
  - cluster name: source/destination
  - cluster management IP: source/destination
  - inter-cluster IP: source/destination
  - vserver name: source/destination
  - vserver management IP: source/destination
  - NFS lifs: source/destination
  - cluster credentials: fsxadmin and vsadmin pwd to be updated in roles/ontap_setup/defaults/main.yml file

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 blank.

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:

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.

```
ansible-playbook -i hosts requirements.yml
```

```
ansible-galaxy collection install -r collections/requirements.yml --force
```

2. Setup target EC2 DB instance.

```
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.

```
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.

```
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.

```
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.

```
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.

```
```
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:
    ### 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:
    # 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.

   `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

   `git clone https://github.com/NetApp-Automation/na_aws_fsx_ec2_deploy.git`

3. Run init to initialize terraform

   `terraform init`

4. Output the execution plan

   `terraform plan -out=main.plan`

5. Apply the execution plan

   `terraform apply "main.plan"`

6. Run destroy to remove the resources when done

   `terraform destroy`
Where to find additional information

To learn more about the NetApp solution automation, review the following website NetApp Solution Automation