# Oracle Database

NetApp Solutions

NetApp
October 20, 2023

# Table of Contents

# Oracle Database

## TR-4981: Active Data Guard Cost Reduction with AWS 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 AWS 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 AWS 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 AWS 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:

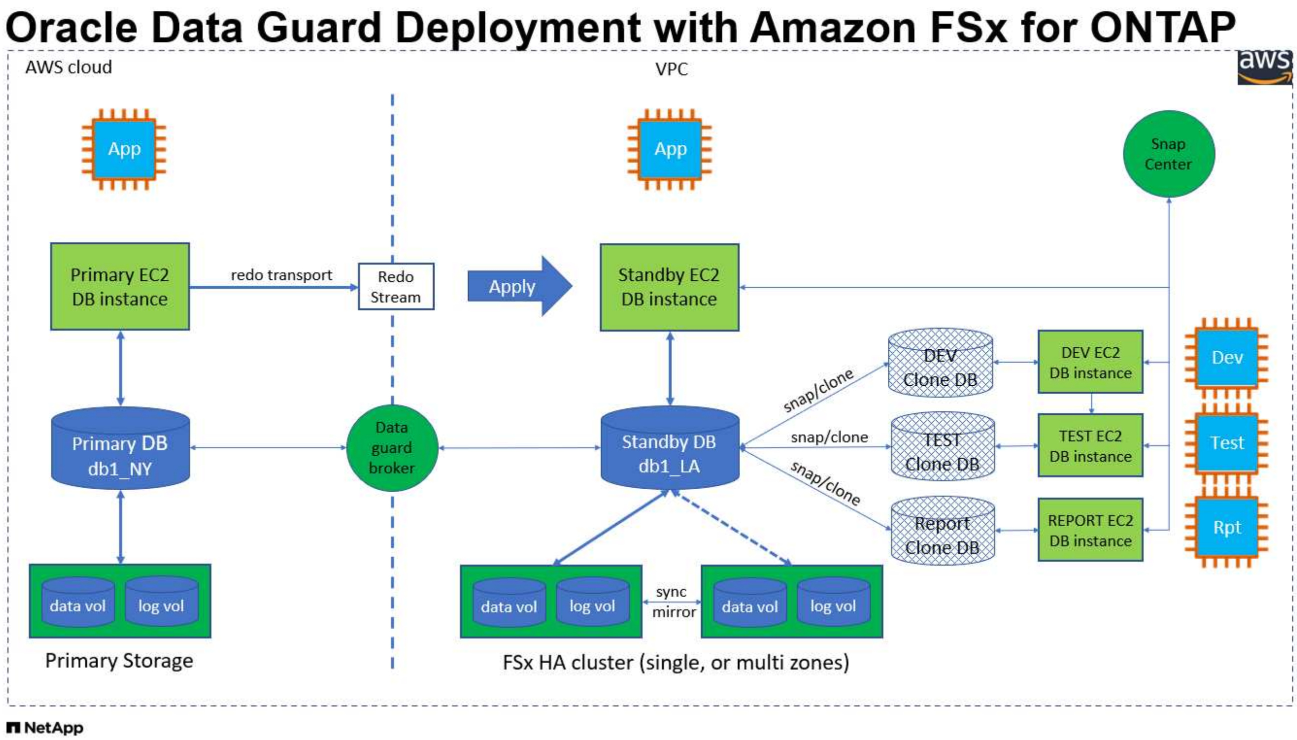
- A DBA who set up Oracle Active Data Guard in AWS for high availability, data protection, and disaster recovery.
- A database solution architect interested in Oracle Active Data Guard configuration in the AWS cloud.
- A storage administrator who manages AWS FSx ONTAP storage that supports Oracle Data 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




### Hardware and software components

| | | |
|---|---|---|
| | **Hardware** | |
| FSx ONTAP storage | Current version offered by AWS | One FSx HA cluster in the same VPC and availability zone |
| EC2 instance for compute | t2.xlarge/4vCPU/16G | Three EC2 T2 xlarge EC2 instances, one as primary DB server, one as standby DB server, and the third as a clone DB server |
| | **Software** | |
| RedHat Linux | RHEL-8.6.0_HVM-20220503-x86_64-2-Hourly2-GP2 | Deployed RedHat subscription for testing |
| 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 |

#### Oracle Data Guard configuration with hypothetical NY to LA DR setup

| Database | DB_UNIQUE_NAME | Oracle Net Service Name |
|---|---|---|
| Primary | db1_NY | db1_NY.demo.netapp.com |
| Physical Standby | db1_LA | db1_LA.demo.netapp.com |

#### Key factors for deployment consideration

- **How Oracle Standby Database FlexClone Works.** AWS FSx ONTAP FlexClone provides shared copies of the same standby database volumes that are writable. The copies of the volumes are actually pointers that link back to original data blocks until a new write initiates on the clone. ONTAP then allocates new storage blocks for the new writes. Any read IOs are serviced by original data blocks under active replication. Thus, the clone are very storage efficient that can be used for many other use cases with minimal and incremental new storage allocation for new write IOs. This provides tremendous storage cost saving by substantially reducing Active Data Guard storage footprint. NetApp recommends to minimize FlexClone activities in the event of database switching over from primary storage to standby FSx storage in order to maintain Oracle performance at high level.
- **Oracle Software Requirements.** In general, a physical standby database must have the same Database Home version as the primary database including Patch Set Exceptions (PSEs), Critical Patch Updates (CPUs), and Patch Set Updates (PSUs), unless an Oracle Data Guard Standby-First Patch Apply process is in progress (as described in My Oracle Support note 1265700.1 at support.oracle.com
- **Standby Database Directory Structure Considerations.** If possible, the data files, log files, and control files on the primary and standby systems should have the same names and path names and use Optimal Flexible Architecture (OFA) naming conventions. The archival directories on the standby database should also be identical between sites, including size and structure. This strategy allows other operations such as backups, switchovers, and failovers to execute the same set of steps, reducing the maintenance complexity.
- **Force Logging Mode.** To protect against unlogged direct writes in the primary database that cannot be propagated to the standby database, turn on FORCE LOGGING at the primary database before performing data file backups for standby creation.
- **Database Storage Management.** For operational simplicity, Oracle recommends that when you set up Oracle Automatic Storage Management (Oracle ASM) and Oracle Managed Files (OMF) in an Oracle Data Guard configuration that you set it up symmetrically on the primary and standby database(s).
- **EC2 compute instances.** In these tests and validations, we used an AWS EC2 t2.xlarge instance as the Oracle database compute instance. NetApp recommends using a M5 type EC2 instance as the compute instance for Oracle in production deployment because it is optimized for database workload. You need to size the EC2 instance appropriately for the number of vCPUs and the amount of RAM based on actual workload requirements.
- **FSx storage HA clusters single- or multi-zone deployment.** In these tests and validations, we deployed an FSx HA cluster in a single AWS availability zone. For production deployment, NetApp recommends deploying an FSx HA pair in two different availability zones. An FSx cluster is alway provisioned in a HA pair that is sync mirrored in a pair of active-passive file systems to provide storage-level redundancy. Multi-zone deployment further enhances high availability in the event of failure in a single AWS zone.
- **FSx storage cluster sizing.** An Amazon FSx for ONTAP storage file system provides up to 160,000 raw SSD IOPS, up to 4GBps throughput, and a maximum of 192TiB capacity. However, you can size the cluster in terms of provisioned IOPS, throughput, and the storage limit (minimum 1,024 GiB) based on your actually requirements at the time of deployment. The capacity can be adjusted dynamically on the fly without affecting application availability.

## Solution deployment

It is assumed that you already have your primary Oracle database deployed in AWS EC2 environment within a VPC as the starting point for setting up Data Guard. The primary database is deployed using Oracle ASM for storage management. Two ASM disk groups - +DATA and +LOGS are created for Oracle data files, log files, and control file etc. For details on Oracle deployment in AWS with ASM, please refer to following technical reports for help.

- Oracle Database Deployment on EC2 and FSx Best Practices
- Oracle Database Deployment and Protection in AWS FSx/EC2 with iSCSI/ASM
- Oracle 19c in Standalone Restart on AWS FSx/EC2 with NFS/ASM

Your primary Oracle database can be running either on an FSx ONTAP or any other storage of choices within the AWS EC2 ecosystem. The following section provides step-by-step deployment procedures for setting up Oracle Data Guard between a primary EC2 DB instance with ASM storage to a standby EC2 DB instance with ASM storage.

### Prerequisites for deployment

Deployment requires the following prerequisites.

1. An AWS account has been set up, and the necessary VPC and network segments have been created within your AWS account.
2. From the AWS EC2 console, you need to deploy minimum three EC2 Linux instances, one as the primary Oracle DB instance, one as standby Oracle DB instance, and an clone target DB instance for reporting, dev, and test etc. See the architecture diagram in the previous section for more details about the environment setup. Also review the AWS User Guide for Linux instances for more information.
3. From the AWS EC2 console, deploy Amazon FSx for ONTAP storage HA clusters to host Oracle volumes that stores the Oracle standby database. If you are not familiar with the deployment of FSx storage, see the documentation Creating FSx for ONTAP file systems for step-by-step instructions.
4. Steps 2 and 3 can be performed using the following Terraform automation toolkit, which creates an EC2 instance named `ora_01` and an FSx file system named `fsx_01`. Review the instruction carefully and change the variables to suit your environment before execution. The template can be easily revised for your own deployment requirements.

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

Ensure that you have allocated at least 50G in EC2 instance root volume in order to have sufficient space to stage Oracle installation files.

### Prepare the primary database for Data Guard

In this demonstration, we have setup a primary Oracle database called db1 on the primary EC2 DB instance with two ASM disk groups in standalone Restart configuration with data files in ASM disk group +DATA and flash recovery area in ASM disk group +LOGS. Following illustrates the detailed procedures for setting up primary database for Data Guard. All steps should be executed as database owner - oracle user.

1. Primary database db1 configuration on primary EC2 DB instance ip-172-30-15-45. The ASM disk groups can be on any type of storage within EC2 ecosystem.

```
[oracle@ip-172-30-15-45 ~]$ cat /etc/oratab

# This file is used by ORACLE utilities.  It is created by root.sh
# and updated by either Database Configuration Assistant while
creating
# a database or ASM Configuration Assistant while creating ASM
instance.

# A colon, ':', is used as the field terminator.  A new line
terminates
# the entry.  Lines beginning with a pound sign, '#', are comments.
#
# Entries are of the form:
#   $ORACLE_SID:$ORACLE_HOME:<N|Y>:
#
# The first and second fields are the system identifier and home
# directory of the database respectively.  The third field indicates
# to the dbstart utility that the database should , "Y", or should
not,
# "N", be brought up at system boot time.
#
# Multiple entries with the same $ORACLE_SID are not allowed.
#
#
+ASM:/u01/app/oracle/product/19.0.0/grid:N
db1:/u01/app/oracle/product/19.0.0/db1:N

[oracle@ip-172-30-15-45 ~]$
/u01/app/oracle/product/19.0.0/grid/bin/crsctl stat res -t
--------------------------------------------------------------------
------------
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/o

racle/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;

    GROUP# TYPE    MEMBER
---------- -------
------------------------------------------------------------
         3 ONLINE  +DATA/DB1/ONLINELOG/group_3.264.1145821513
         2 ONLINE  +DATA/DB1/ONLINELOG/group_2.263.1145821513
         1 ONLINE  +DATA/DB1/ONLINELOG/group_1.262.1145821513
         4 STANDBY +DATA/DB1/ONLINELOG/group_4.286.1146082751
         4 STANDBY +LOGS/DB1/ONLINELOG/group_4.258.1146082753
         5 STANDBY +DATA/DB1/ONLINELOG/group_5.287.1146082819
         5 STANDBY +LOGS/DB1/ONLINELOG/group_5.260.1146082821
         6 STANDBY +DATA/DB1/ONLINELOG/group_6.288.1146082825
         6 STANDBY +LOGS/DB1/ONLINELOG/group_6.261.1146082827
         7 STANDBY +DATA/DB1/ONLINELOG/group_7.289.1146082835
         7 STANDBY +LOGS/DB1/ONLINELOG/group_7.262.1146082835

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.

```
DB_NAME=db1
DB_UNIQUE_NAME=db1_NY
LOG_ARCHIVE_CONFIG='DG_CONFIG=(db1_NY,db1_LA)'
LOG_ARCHIVE_DEST_1='LOCATION=USE_DB_RECOVERY_FILE_DEST
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=db1_NY'
LOG_ARCHIVE_DEST_2='SERVICE=db1_LA ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=db1_LA'
REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE
FAL_SERVER=db1_LA
STANDBY_FILE_MANAGEMENT=AUTO
```

8. From sqlplus, create spfile in ASM +DATA directory from revised pfile in /home/oracle directory.

```
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.

```
[oracle@ip-172-30-15-45 db1]$ srvctl config database -d db1
Database unique name: db1
Database name: db1
Oracle home: /u01/app/oracle/product/19.0.0/db1
Oracle user: oracle
Spfile: +DATA/DB1/PARAMETERFILE/spfile.270.1145822903
Password file:
Domain: demo.netapp.com
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Disk Groups: DATA
Services:
OSDBA group:
OSOPER group:
Database instance: db1
[oracle@ip-172-30-15-45 db1]$ srvctl modify database -d db1 -spfile
+DATA/DB1/PARAMETERFILE/spfiledb1.ora
[oracle@ip-172-30-15-45 db1]$ srvctl config database -d db1
Database unique name: db1
Database name: db1
Oracle home: /u01/app/oracle/product/19.0.0/db1
Oracle user: oracle
Spfile: +DATA/DB1/PARAMETERFILE/spfiledb1.ora
Password file:
Domain: demo.netapp.com
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Disk Groups: DATA
Services:
OSDBA group:
OSOPER group:
Database instance: db1
```

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.

   ```
   scp
   oracle@172.30.15.45:/u01/app/oracle/product/19.0.0/db1/dbs/orapwdb1
   .
   ```

4. Create tnsnames.ora file with following entries.

```
# 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)
    )
  )
```

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.

```
[oracle@ip-172-30-15-67 bin]$ dbca -silent -createDuplicateDB
-gdbName db1 -primaryDBConnectionString ip-172-30-15-
45.ec2.internal:1521/db1_NY.demo.netapp.com -sid db1 -initParams
fal_server=db1_NY -createAsStandby -dbUniqueName db1_LA
Enter SYS user password:

Prepare for db operation
22% complete
Listener config step
44% complete
Auxiliary instance creation
67% complete
RMAN duplicate
89% complete
Post duplicate database operations
100% complete

Look at the log file
"/u01/app/oracle/cfgtoollogs/dbca/db1_LA/db1_LA.log" for further
details.
```

8. Validate duplicated standby database. Newly duplicated standby database open in READ ONLY mode initially.

```
[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;

NAME      OPEN_MODE
--------- --------------------
DB1       READ ONLY
```

```
SQL> show parameter name

NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
cdb_cluster_name                     string
cell_offloadgroup_name               string
db_file_name_convert                 string
db_name                              string      db1
db_unique_name                       string      db1_LA
global_names                         boolean     FALSE
instance_name                        string      db1
lock_name_space                      string
log_file_name_convert                string
pdb_file_name_convert                string
processor_group_name                 string

NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
service_names                        string
db1_LA.demo.netapp.com
SQL>
SQL> show parameter log_archive_config

NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
log_archive_config                   string
DG_CONFIG=(db1_NY,db1_LA)
SQL> show parameter fal_server

NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
fal_server                           string      db1_NY

SQL> select name from v$datafile;

NAME
--------------------------------------------------------------------
------------
+DATA/DB1_LA/DATAFILE/system.261.1146248215
+DATA/DB1_LA/DATAFILE/sysaux.262.1146248231
+DATA/DB1_LA/DATAFILE/undotbs1.263.1146248247
+DATA/DB1_LA/03C5C01A66EE9797E0632D0F1EAC5F59/DATAFILE/system.264.11
```

```
46248253
+DATA/DB1_LA/03C5C01A66EE9797E0632D0F1EAC5F59/DATAFILE/sysaux.265.11
46248261
+DATA/DB1_LA/DATAFILE/users.266.1146248267
+DATA/DB1_LA/03C5C01A66EE9797E0632D0F1EAC5F59/DATAFILE/undotbs1.267.
1146248269
+DATA/DB1_LA/03C5EFD07C41A1FAE0632D0F1EAC9BD8/DATAFILE/system.268.11
46248271
+DATA/DB1_LA/03C5EFD07C41A1FAE0632D0F1EAC9BD8/DATAFILE/sysaux.269.11
46248279
+DATA/DB1_LA/03C5EFD07C41A1FAE0632D0F1EAC9BD8/DATAFILE/undotbs1.270.
1146248285
+DATA/DB1_LA/03C5EFD07C41A1FAE0632D0F1EAC9BD8/DATAFILE/users.271.114
6248293

NAME
--------------------------------------------------------------------
------------
+DATA/DB1_LA/03C5F0DDF35CA2B6E0632D0F1EAC8B6B/DATAFILE/system.272.11
46248295
+DATA/DB1_LA/03C5F0DDF35CA2B6E0632D0F1EAC8B6B/DATAFILE/sysaux.273.11
46248301
+DATA/DB1_LA/03C5F0DDF35CA2B6E0632D0F1EAC8B6B/DATAFILE/undotbs1.274.
1146248309
+DATA/DB1_LA/03C5F0DDF35CA2B6E0632D0F1EAC8B6B/DATAFILE/users.275.114
6248315
+DATA/DB1_LA/03C5F1C9B142A2F1E0632D0F1EACF21A/DATAFILE/system.276.11
46248317
+DATA/DB1_LA/03C5F1C9B142A2F1E0632D0F1EACF21A/DATAFILE/sysaux.277.11
46248323
+DATA/DB1_LA/03C5F1C9B142A2F1E0632D0F1EACF21A/DATAFILE/undotbs1.278.
1146248331
+DATA/DB1_LA/03C5F1C9B142A2F1E0632D0F1EACF21A/DATAFILE/users.279.114
6248337

19 rows selected.

SQL> select name from v$controlfile;

NAME
--------------------------------------------------------------------
------------
+DATA/DB1_LA/CONTROLFILE/current.260.1146248209
+LOGS/DB1_LA/CONTROLFILE/current.257.1146248209

SQL> select name from v$tempfile;
```

```
NAME
--------------------------------------------------------------------
------------
+DATA/DB1_LA/TEMPFILE/temp.287.1146248371
+DATA/DB1_LA/03C5C01A66EE9797E0632D0F1EAC5F59/TEMPFILE/temp.288.1146
248375
+DATA/DB1_LA/03C5EFD07C41A1FAE0632D0F1EAC9BD8/TEMPFILE/temp.290.1146
248463
+DATA/DB1_LA/03C5F0DDF35CA2B6E0632D0F1EAC8B6B/TEMPFILE/temp.291.1146
248463
+DATA/DB1_LA/03C5F1C9B142A2F1E0632D0F1EACF21A/TEMPFILE/temp.292.1146
248463

SQL> select group#, type, member from v$logfile order by 2, 1;

    GROUP# TYPE    MEMBER
---------- -------
------------------------------------------------------------
         1 ONLINE  +LOGS/DB1_LA/ONLINELOG/group_1.259.1146248349
         1 ONLINE  +DATA/DB1_LA/ONLINELOG/group_1.280.1146248347
         2 ONLINE  +DATA/DB1_LA/ONLINELOG/group_2.281.1146248351
         2 ONLINE  +LOGS/DB1_LA/ONLINELOG/group_2.258.1146248353
         3 ONLINE  +DATA/DB1_LA/ONLINELOG/group_3.282.1146248355
         3 ONLINE  +LOGS/DB1_LA/ONLINELOG/group_3.260.1146248355
         4 STANDBY +DATA/DB1_LA/ONLINELOG/group_4.283.1146248357
         4 STANDBY +LOGS/DB1_LA/ONLINELOG/group_4.261.1146248359
         5 STANDBY +DATA/DB1_LA/ONLINELOG/group_5.284.1146248361
         5 STANDBY +LOGS/DB1_LA/ONLINELOG/group_5.262.1146248363
         6 STANDBY +LOGS/DB1_LA/ONLINELOG/group_6.263.1146248365
         6 STANDBY +DATA/DB1_LA/ONLINELOG/group_6.285.1146248365
         7 STANDBY +LOGS/DB1_LA/ONLINELOG/group_7.264.1146248369
         7 STANDBY +DATA/DB1_LA/ONLINELOG/group_7.286.1146248367

14 rows selected.

SQL> select name, open_mode from v$database;
```

```
NAME      OPEN_MODE
--------- --------------------
DB1       READ ONLY
```

9. Restart standby database in `mount` stage and execute following command to activate standby database managed recovery.

```
alter database recover managed standby database disconnect from
session;
```

```
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.

Total System Global Area 8053062944 bytes
Fixed Size                    9182496 bytes
Variable Size              1291845632 bytes
Database Buffers           6744440832 bytes
Redo Buffers                  7593984 bytes
Database mounted.
SQL> alter database recover managed standby database disconnect from
session;

Database altered.
```

10. Validate the standby database recovery status. Notice the `recovery logmerger` in `APPLYING_LOG` action.

```
SQL> SELECT ROLE, THREAD#, SEQUENCE#, ACTION FROM
V$DATAGUARD_PROCESS;

ROLE                        THREAD#  SEQUENCE# ACTION
------------------------ ---------- ---------- ------------
recovery apply slave              0          0 IDLE
recovery apply slave              0          0 IDLE
recovery apply slave              0          0 IDLE
recovery apply slave              0          0 IDLE
recovery logmerger                1         30 APPLYING_LOG
RFS ping                          1         30 IDLE
RFS async                         1         30 IDLE
archive redo                      0          0 IDLE
archive redo                      0          0 IDLE
archive redo                      0          0 IDLE
gap manager                       0          0 IDLE

ROLE                        THREAD#  SEQUENCE# ACTION
------------------------ ---------- ---------- ------------
managed recovery                  0          0 IDLE
redo transport monitor            0          0 IDLE
log writer                        0          0 IDLE
archive local                     0          0 IDLE
redo transport timer              0          0 IDLE

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 databse for other use cases

The key benefit of staging standby database on AWS FSx ONTAP in Data Guard is that it can be FlexCloned to serve many other use cases with minimal additional storage investment. In the following section, we demonstrate how to snapshot and clone the mounted and under recovery standby database volumes on FSx ONTAP for other purposes, such as DEV, TEST, REPORT, etc., using the NetApp SnapCenter tool.

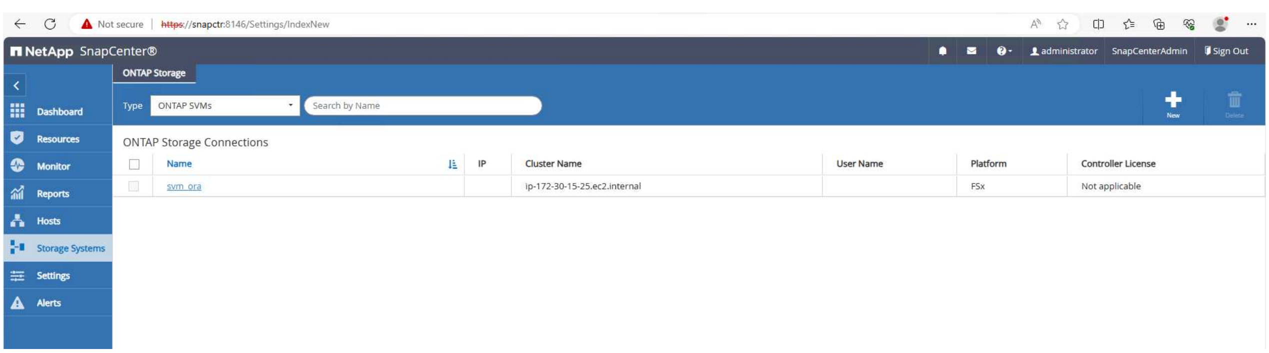
Following are high level procedures to clone a READ/WRITE database from the managed physical standby database in Data Guard using SnapCenter. For detail instructions on how to setup and configure SnapCenter, please refer to Hybrid Cloud Database Solutions with SnapCenter relavant Oracle sections.

1. We begin with creating a test table and inserting a row into the test table on primary database. We will then validate if the transaction traverse down to standby and finally the clone.

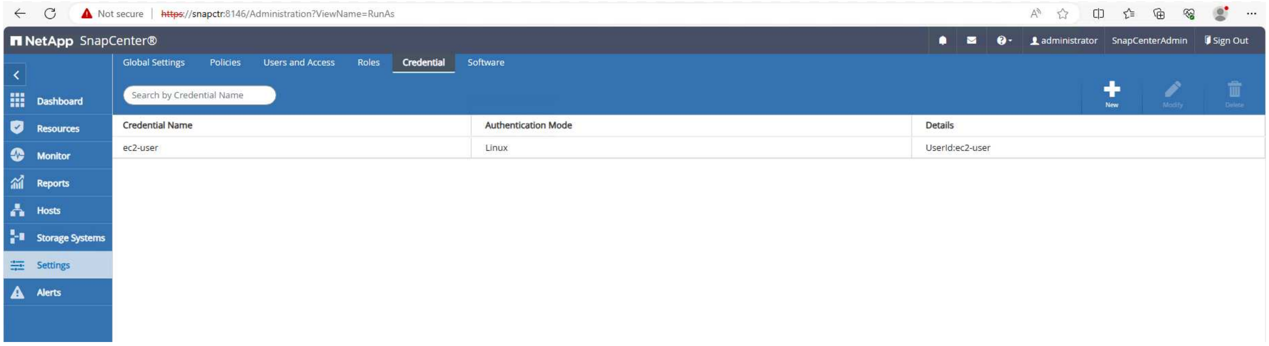
```
[oracle@ip-172-30-15-45 db1]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Aug 31 16:35:53
2023
Version 19.18.0.0.0

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


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0

SQL> alter session set container=db1_pdb1;

Session altered.

SQL> create table test(
  2  id integer,
  3  dt timestamp,
  4  event varchar(100));

Table created.

SQL> insert into test values(1, sysdate, 'a test transaction on
primary database db1 and ec2 db host: ip-172-30-15-
45.ec2.internal');

1 row created.

SQL> commit;

Commit complete.
```

```
SQL> select * from test;

        ID
----------
DT
---------------------------------------------------------------------
-------
EVENT
---------------------------------------------------------------------
------------
         1
31-AUG-23 04.49.29.000000 PM
a test transaction on primary database db1 and ec2 db host: ip-172-
30-15-45.ec2.
internal

SQL> select instance_name, host_name from v$instance;

INSTANCE_NAME
----------------
HOST_NAME
----------------------------------------------------------------
db1
ip-172-30-15-45.ec2.internal
```

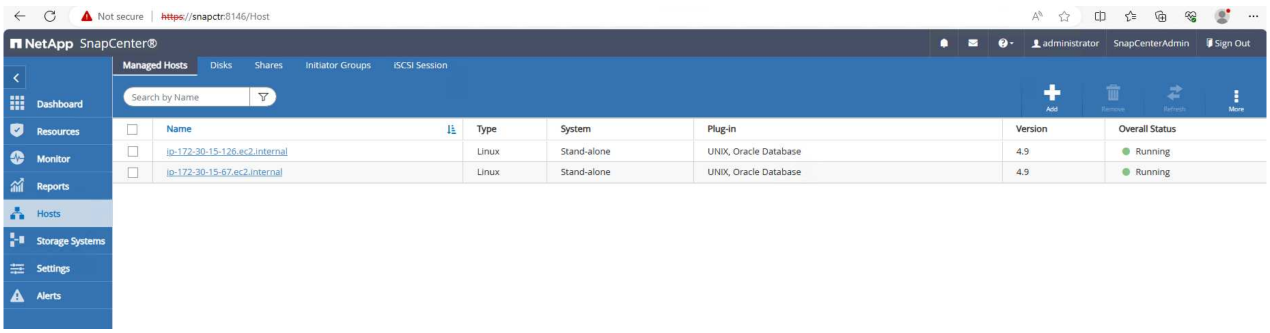
2. Add FSx storage cluster to `Storage Systems` in SnapCenter with FSx cluster management IP and fsxadmin credential.



3. Add AWS ec2-user to `Credential` in `Settings`.

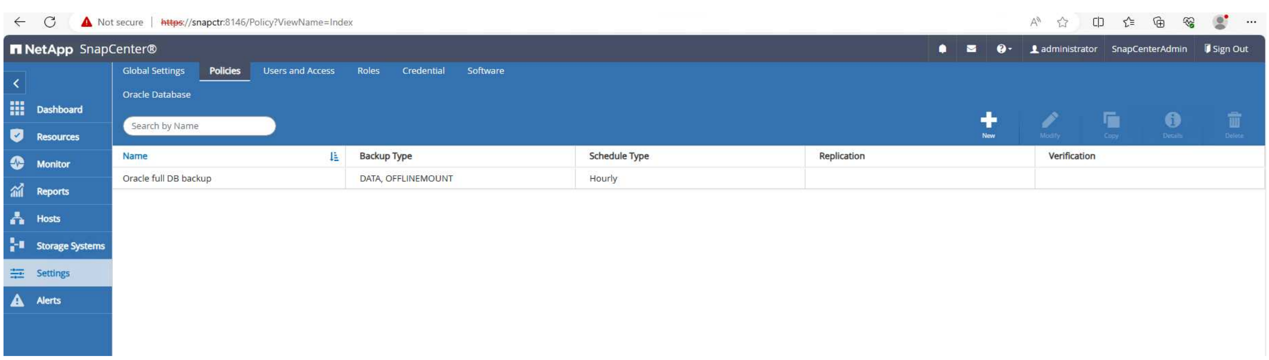



4. Add standby EC2 DB instance and clone EC2 DB instance to `Hosts`.

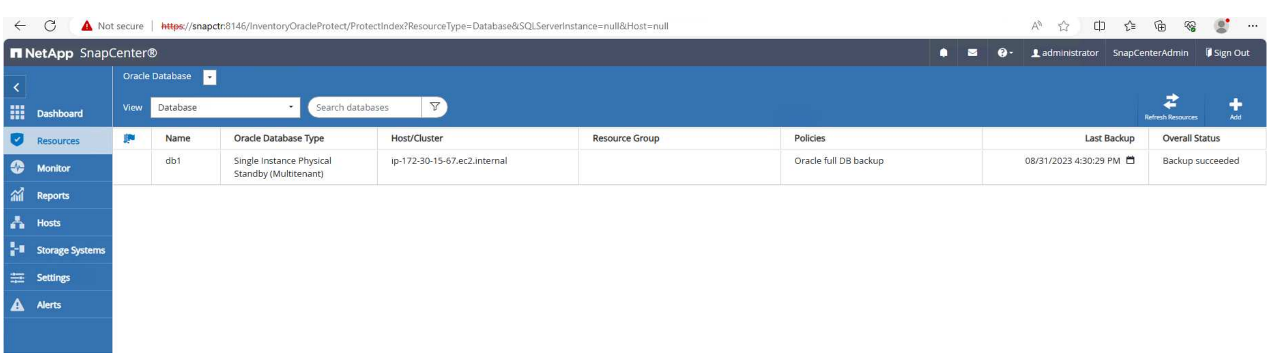



The clone EC2 DB instance should have similar Oracle software stacks installed and configured. In our test case, the grid infrastructure and Oracle 19C installed and configured but no database created.

5. Create a backup policy that is tailored for offline/mount full database backup.

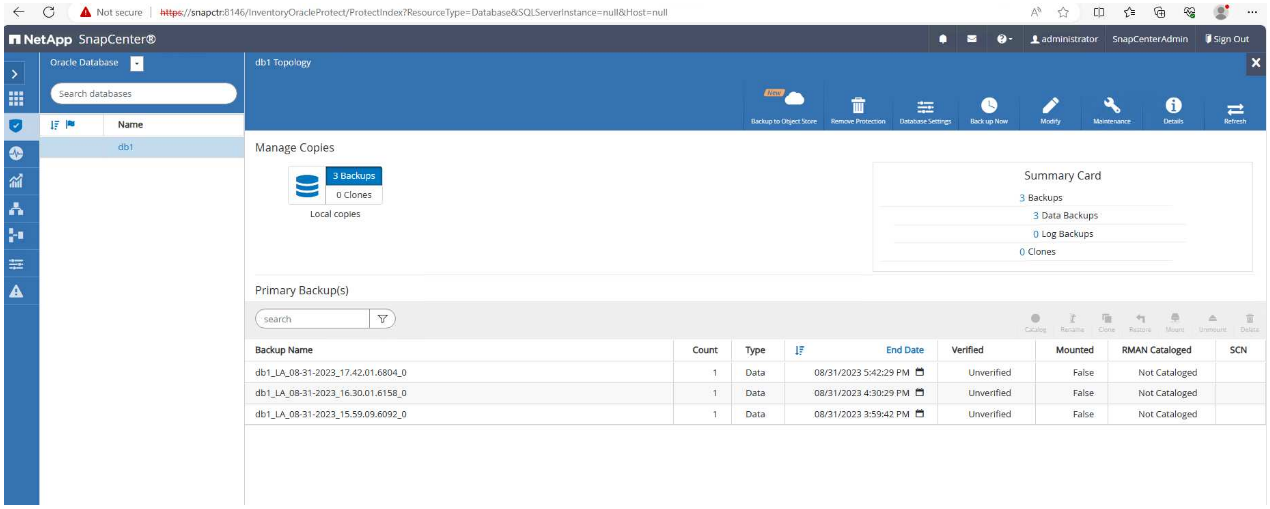



6. Apply backup policy to protect standby database in `Resources` tab.

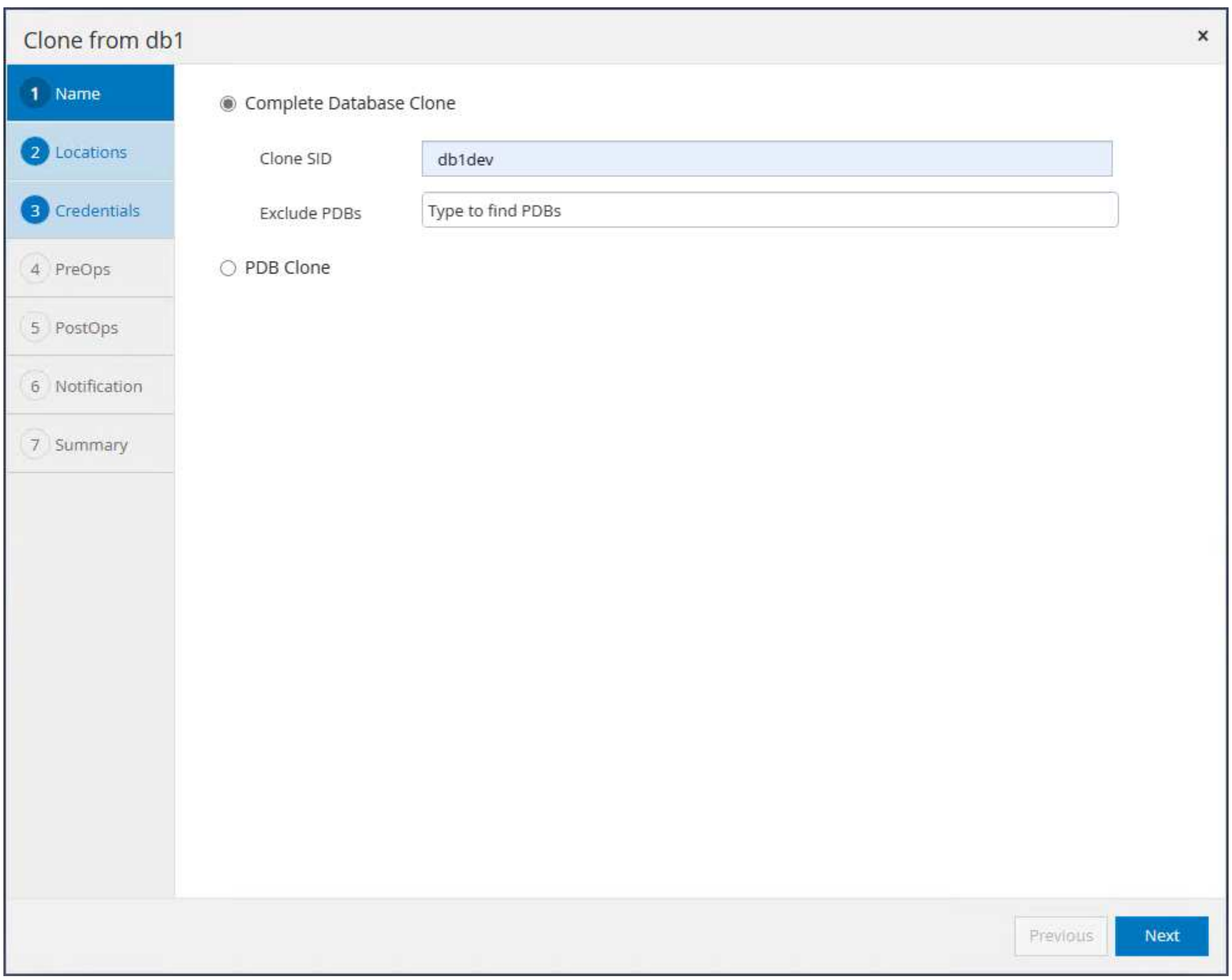



7. Click on database name to open the database backups page. Select a backup to be used for

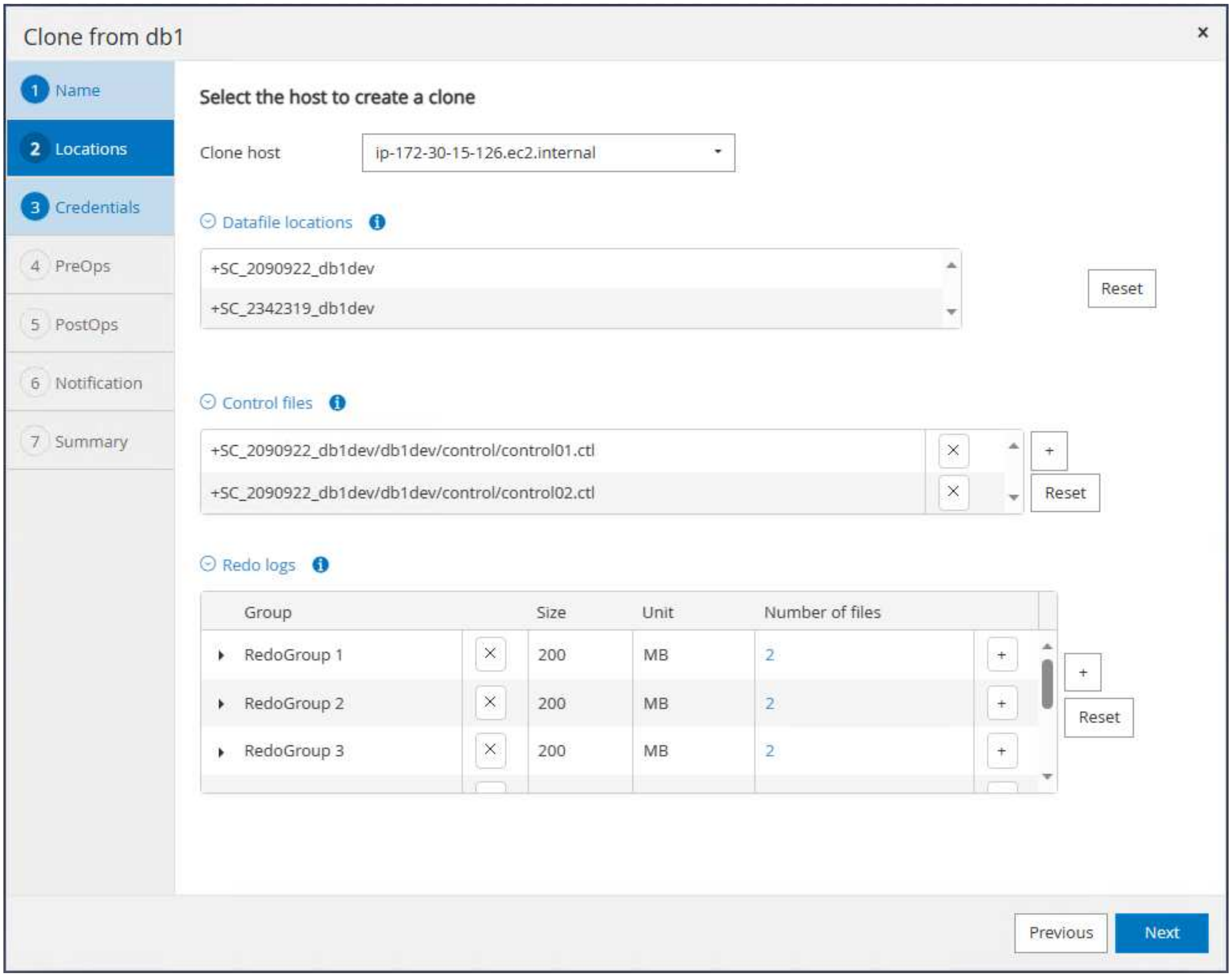
database clone and click on `Clone` button to launch clone workflow.



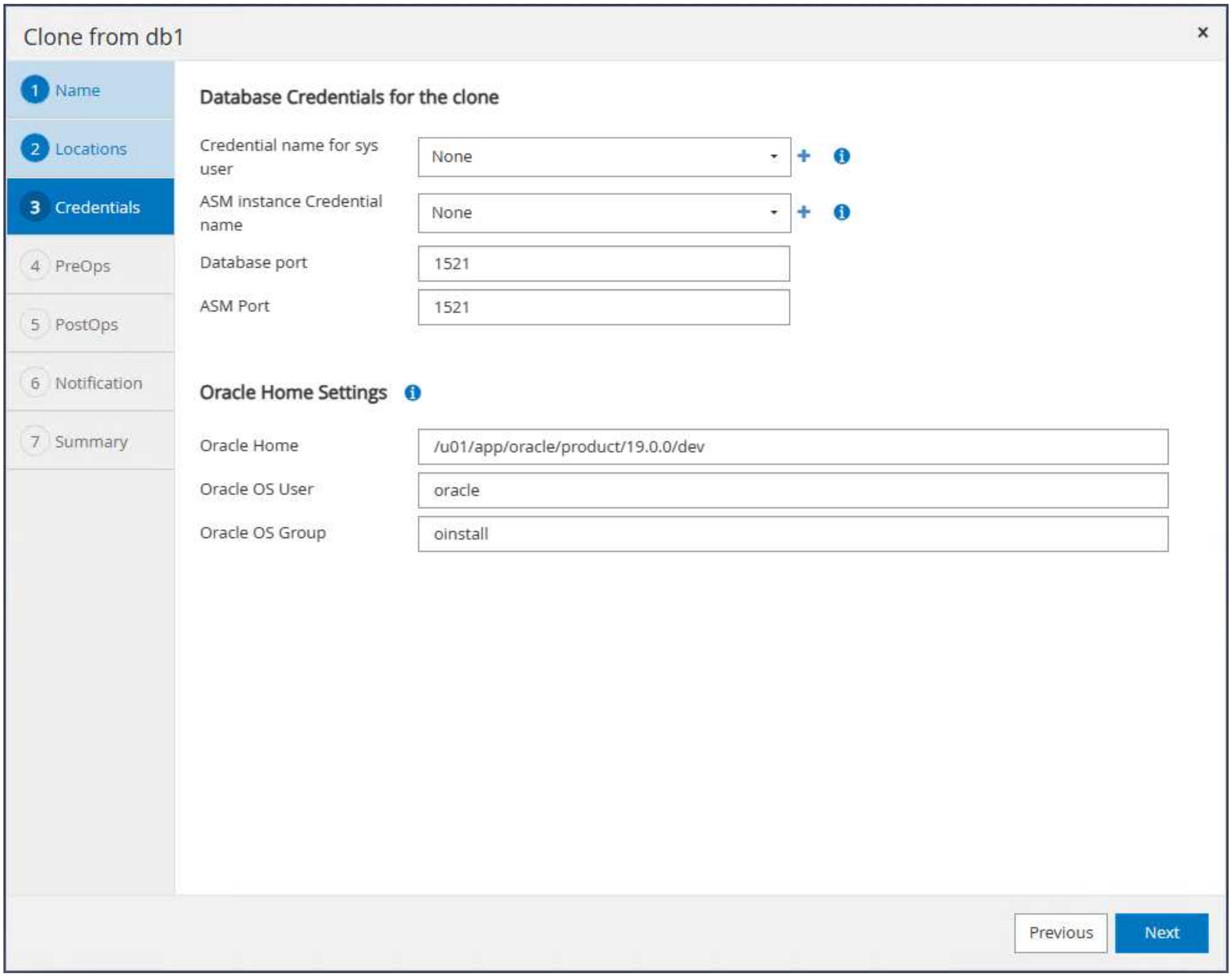
8. Select `Complete Database Clone` and name the clone instance SID.



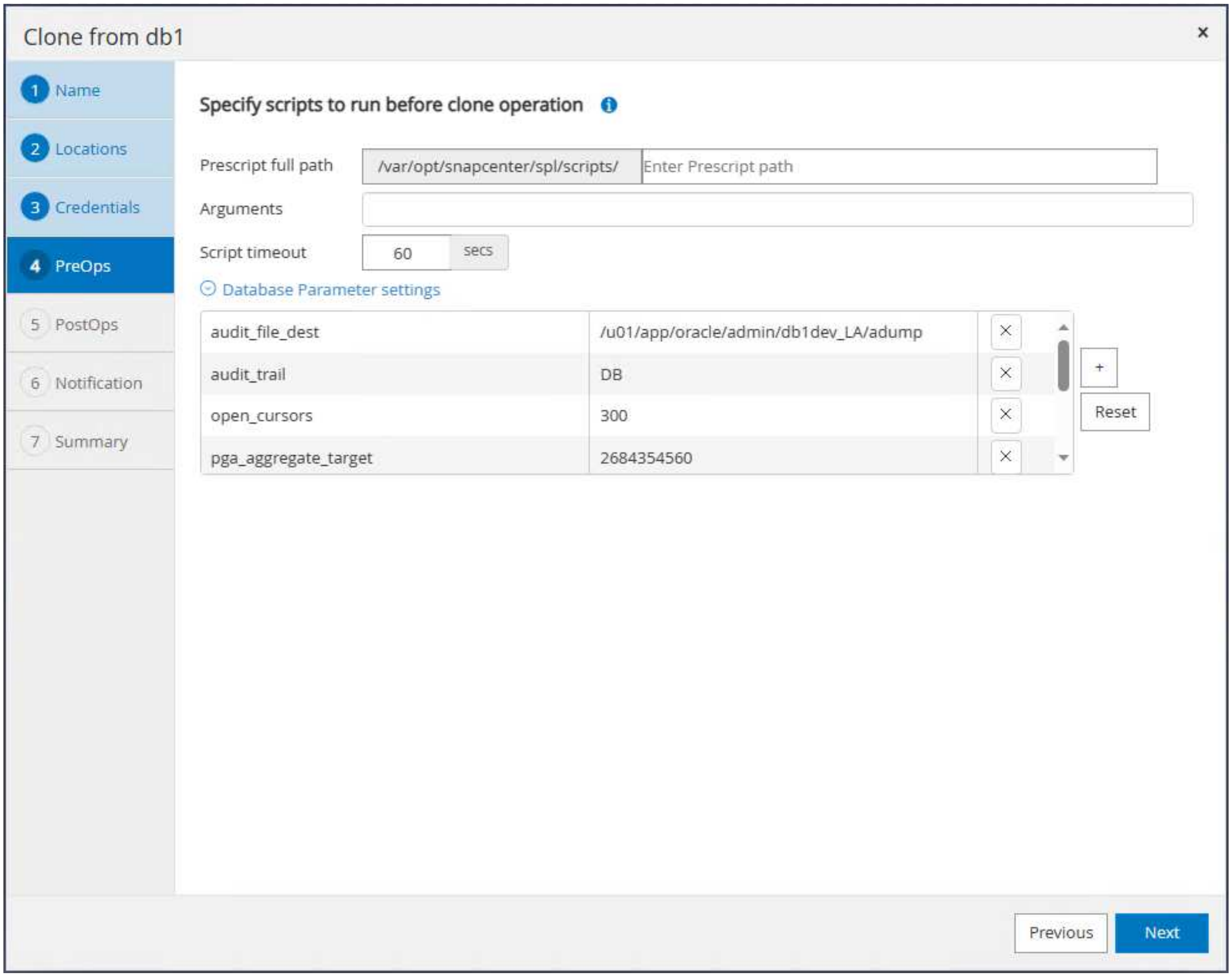

9. Select the clone host, which hosts the cloned database from standby DB. Accept the default for data files, control files, and redo logs. Two ASM disk groups will be created on the clone host that are corresponding to the disk groups on standby database.



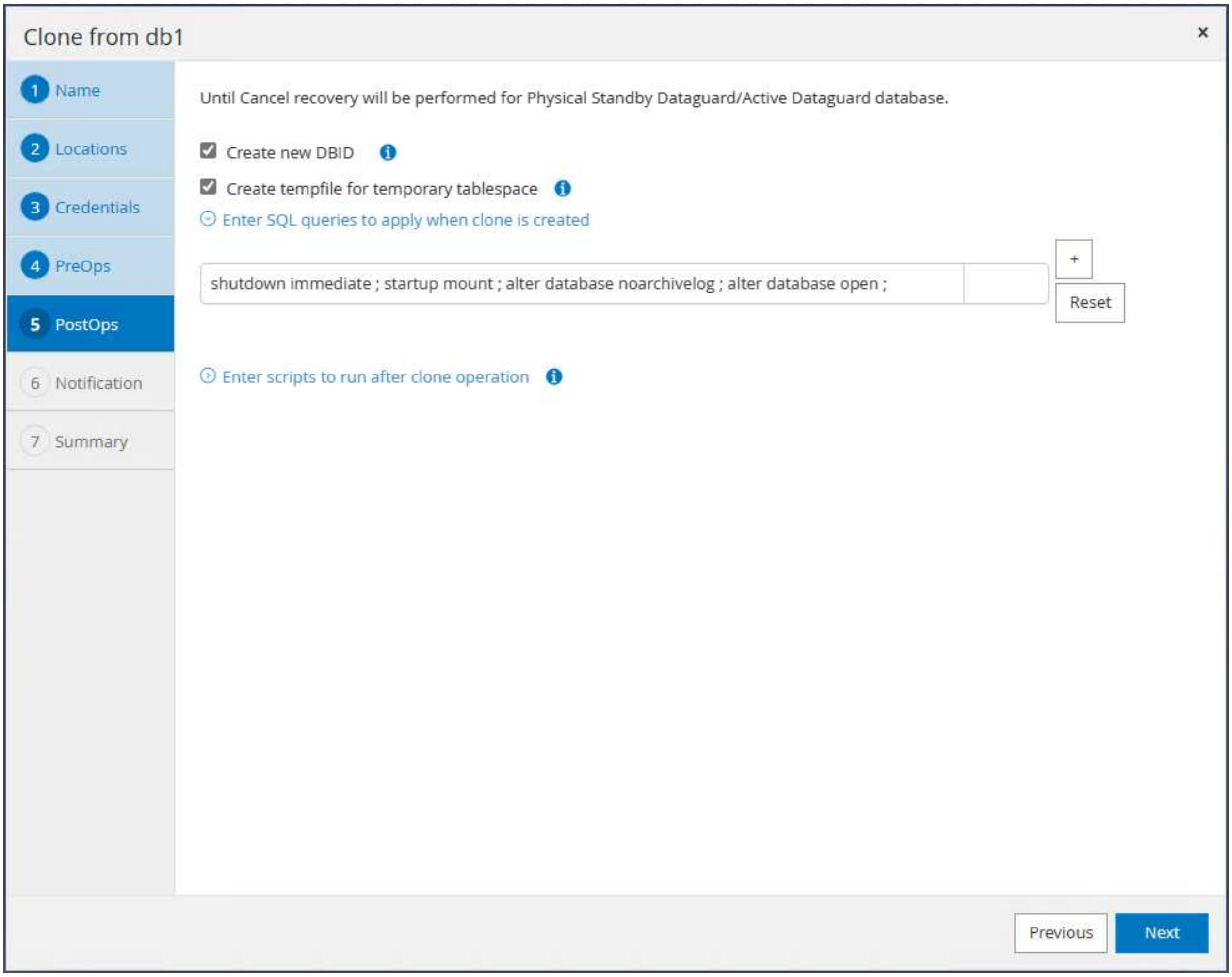

10. No database credentials are needed for OS based authentication. Match Oracle home setting with what is configured on the clone EC2 database instance.



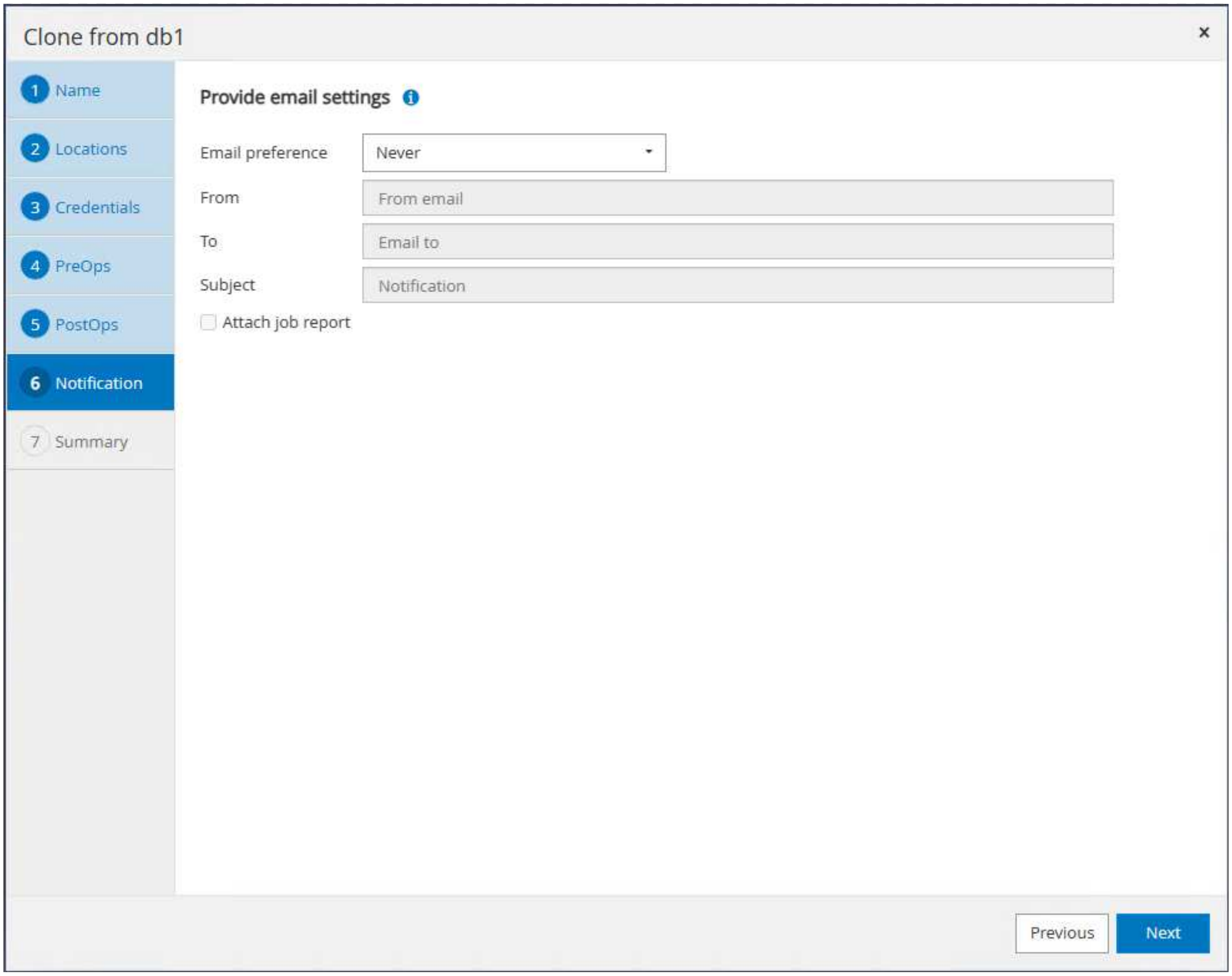

11. Change clone database parameters if needed and specify scripts to run before cloen if any.



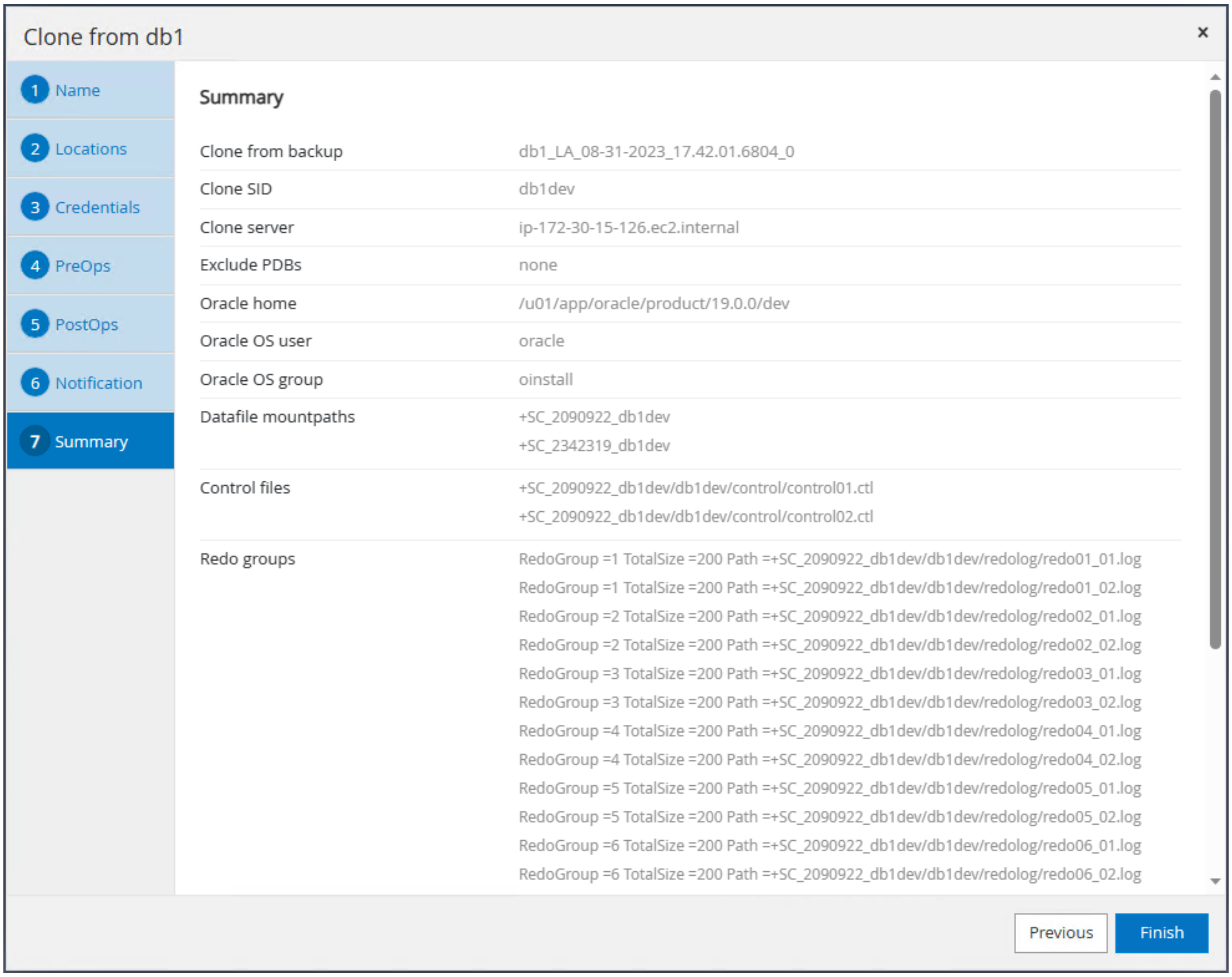

12. Enter SQL to run after clone. In the demo, we executed commands to turn off database archive mode for a dev/test/report database.



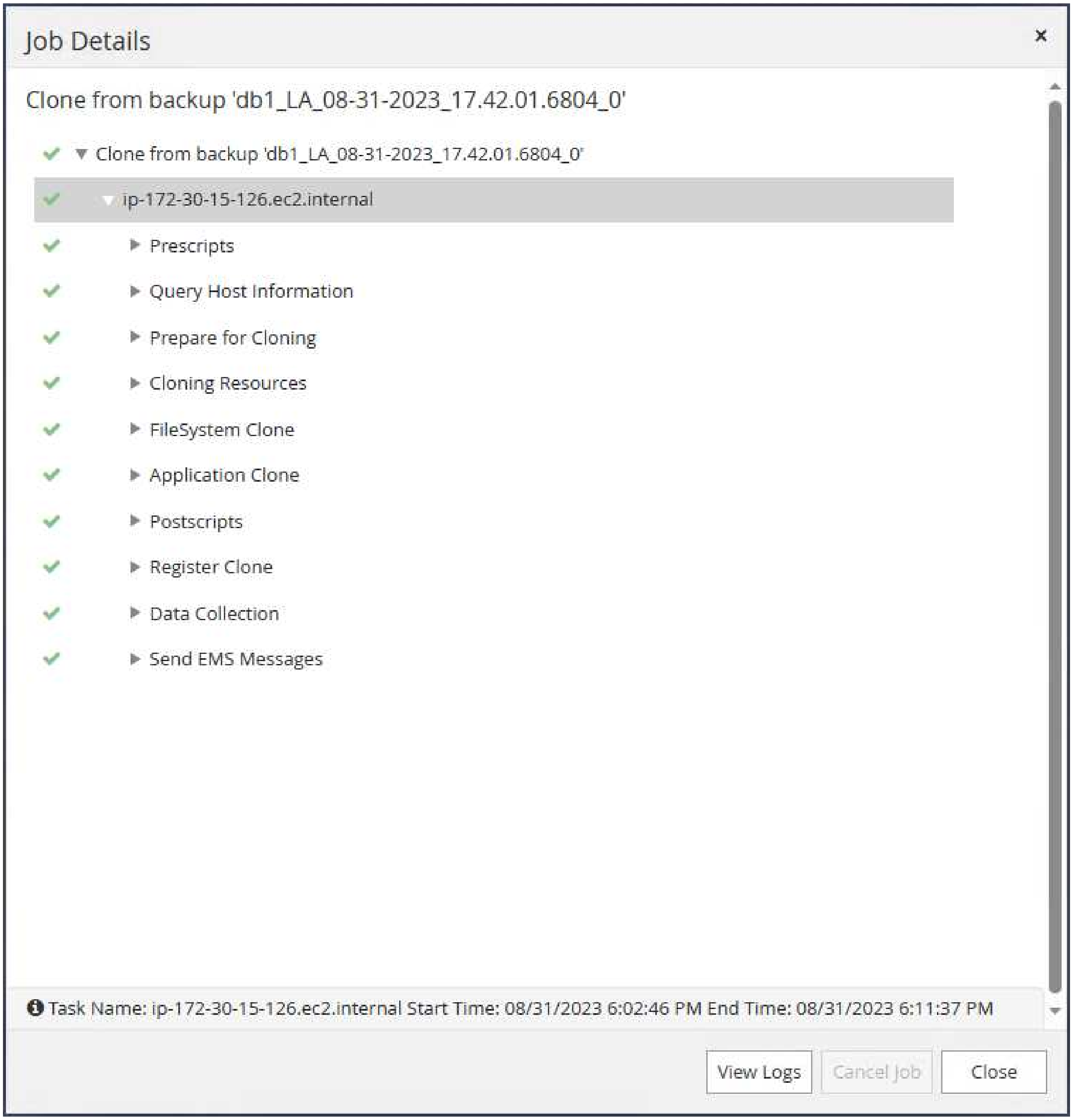

13. Configure email notification if desired.



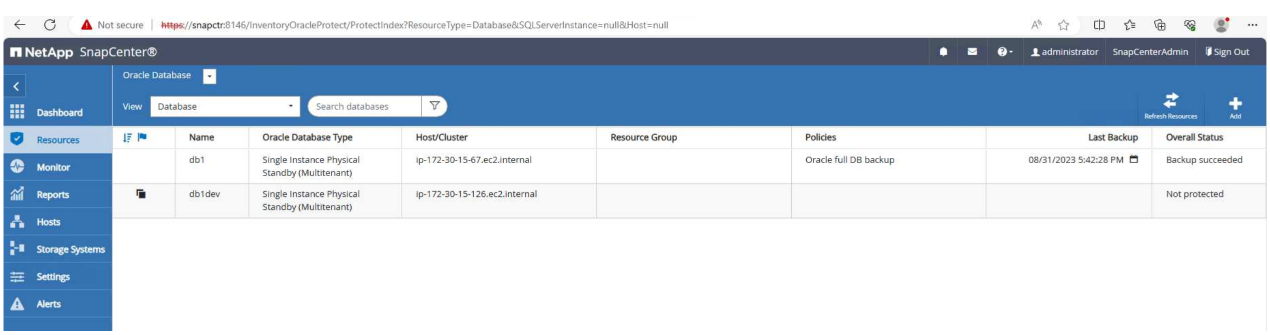

14. Review the summary, click `Finish` to start the clone.




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;

        ID
----------
DT
--------------------------------------------------------------------
-------
EVENT
--------------------------------------------------------------------
------------
```

```
          1
31-AUG-23 04.49.29.000000 PM
a test transaction on primary database db1 and ec2 db host: ip-172-
30-15-45.ec2.
internal


SQL>
```

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

  https://docs.netapp.com/us-en/netapp-solutions/databases/aws_ora_fsx_ec2_deploy_intro.html

- 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!467723097970!e!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARIsAJzSrdqwQrghn6I71jiWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:G:s&s_kwcid=AL!4422!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

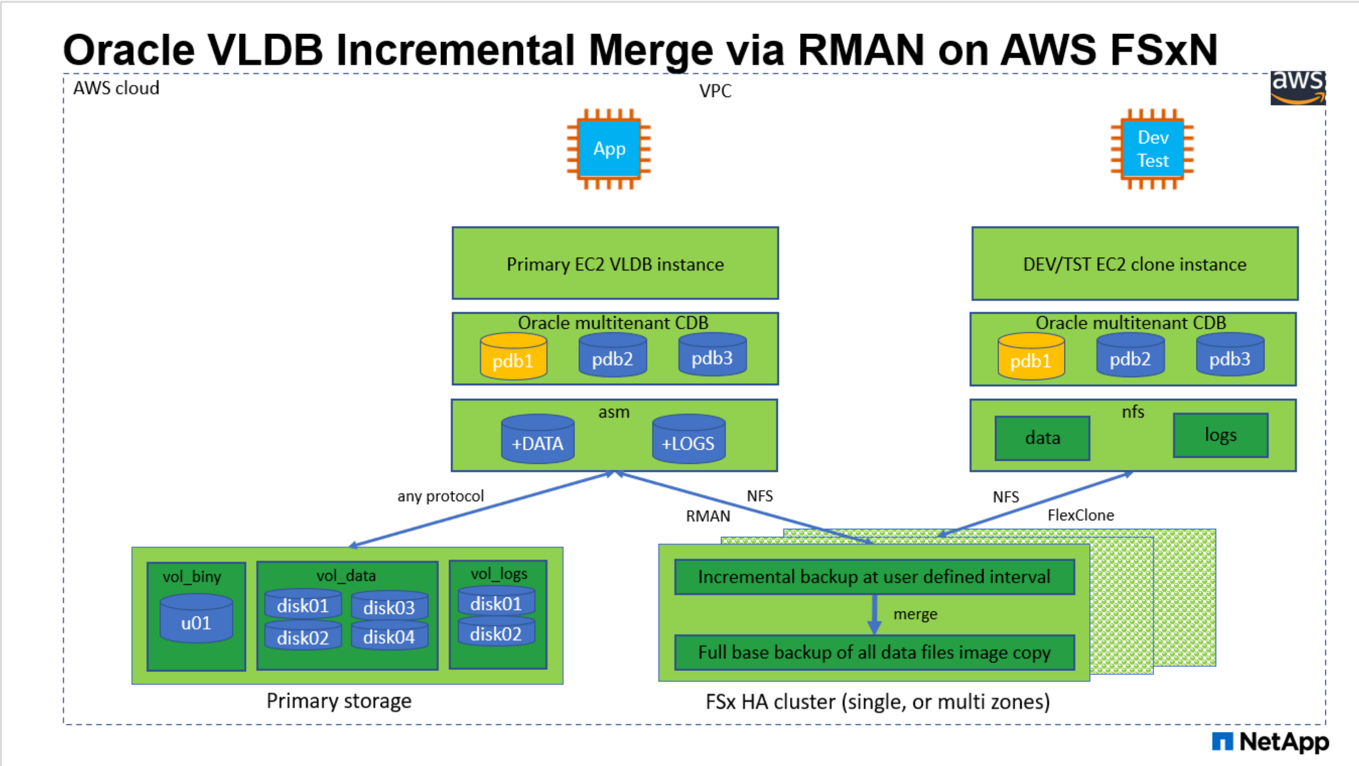
This solution is intended for the following people:

- 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 | |
|---|---|---|
| FSx ONTAP storage | Current version offered by AWS | One FSx HA cluster in the same VPC and availability zone |
| EC2 instance for compute | t2.xlarge/4vCPU/16G | Two EC2 T2 xlarge EC2 instances, one as primary DB server and the other as a clone DB server |
| | **Software** | |
| RedHat Linux | RHEL-8.6.0_HVM-20220503-x86_64-2-Hourly2-GP2 | Deployed RedHat subscription for testing |
| 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 |

## 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.

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

> Ensure that you have allocated at least 50G in EC2 instance root volume in order to have sufficient space to stage Oracle installation files.

#### Provision and export NFS volume to be mounted to EC2 DB instance host

In this demonstration, we will show how to provision an NFS volume from the command line by login to an FSx cluster via ssh as fsxadmin user through FSx cluster management IP. Alternatively, the volume can be allocated using the AWS FSx console as well. Repeat the procedures on other FSx file systems if more than one FSx file system are set up to accommodate the size of the database.

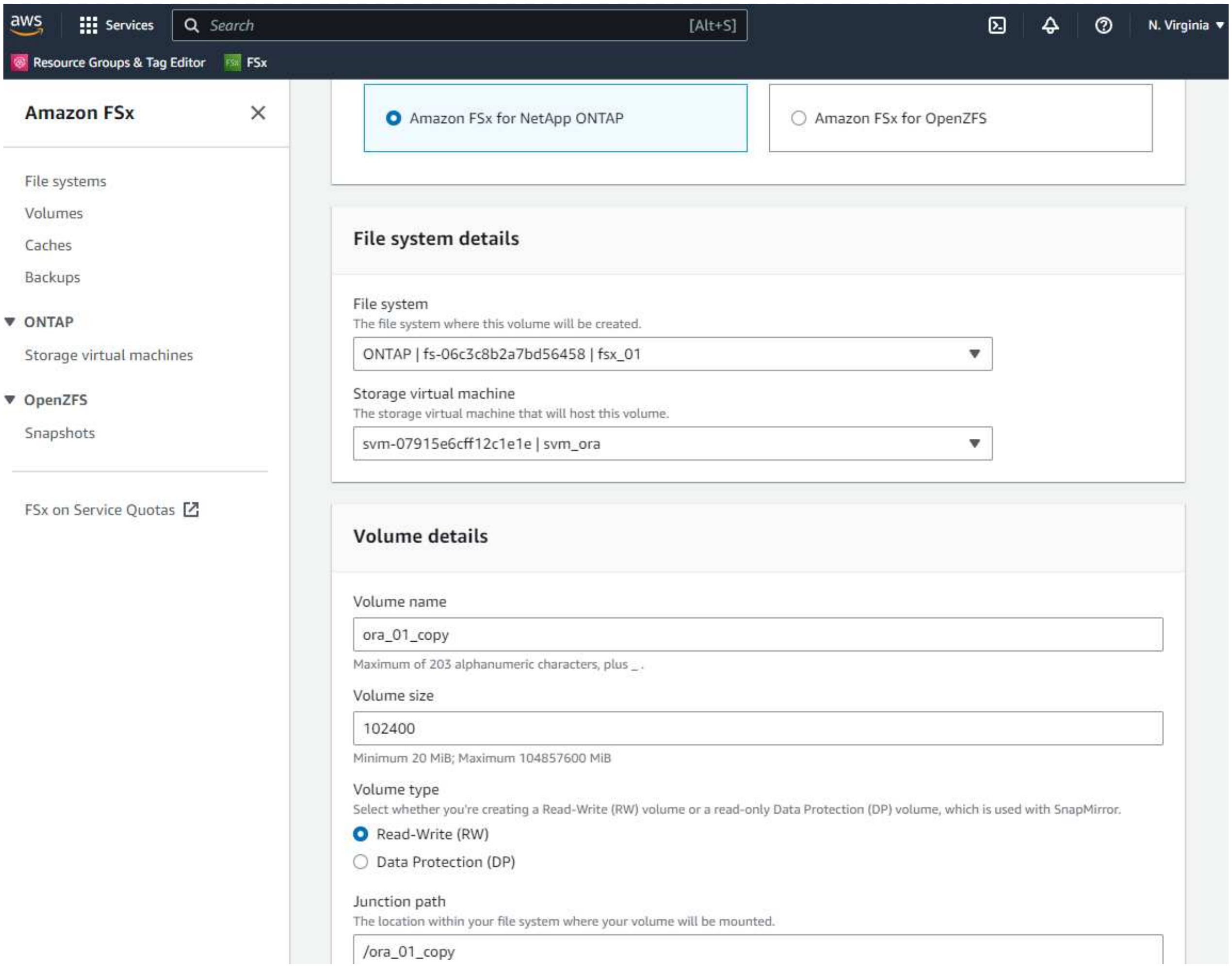
1. First, provision NFS volume via CLI by logging to the FSx cluster through SSH as the fsxadmin user. Change to your FSx cluster management IP address, which can be retrieved from AWS FSx ONTAP UI console.

   ```
   ssh fsxadmin@172.30.15.53
   ```

2. Create NFS volume the same size as your primary storage for storing primary Oracle VLDB database data files image copy.

   ```
   vol create -volume ora_01_copy -aggregate aggr1 -size 100G -state
   online -type RW -junction-path /ora_01_copy -snapshot-policy none
   -tiering-policy snapshot-only
   ```

3. Alternatively, the volume can be provisioned from AWS FSx console UI with options: storage efficiency `Enabled`, security style `Unix` , Snapshot policy `None`, and Storage tiering `Snapshot Only` as show below.




4. Create a customized snapshot policy for oracle database with a daily schedule and 30 days retention. You should adjust the policy to fit your specific needs in terms of snapshot frequency and retention window.

```
snapshot policy create -policy oracle -enabled true -schedule1 daily
-count1 30
```

Apply policy to provisioned NFS volume for RMAN incremental backup and merge.

```
vol modify -volume ora_01_copy -snapshot-policy oracle
```

5. Login to EC2 instance as ec2-user and create a directory /nfsfsxn. Create additional mount point directories for additional FSx file systems.

```
sudo mkdir /nfsfsxn
```

6. Mount the FSx ONTAP NFS volume to EC2 DB instance host. Change to your FSx virtual server NFS lif address. The NFS lif address can be retrieved from FSx ONTAP UI console.

```
sudo mount 172.30.15.19:/ora_01_copy /nfsfsxn -o
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=262144,wsize=262144,noin
tr
```

7. Change mount point ownership to oracle:oisntall, 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.

```
rman target / nocatalog;

output:

[oracle@ip-172-30-15-99 ~]$ rman target / nocatalog;

Recovery Manager: Release 19.0.0.0.0 - Production on Wed May 24
17:44:49 2023
Version 19.18.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights
reserved.

connected to target database: DB1 (DBID=1730530050)
using target database control file instead of recovery catalog

RMAN>
```

6. From RMAN prompt, execute the script. First execution creates a baseline database image copy and subsequent executions merge and update the baseline image copy incrementally. The following is how to execute the script and the typical output. Set the number of channels to match the CPU cores on the host.

```
RMAN> @/home/oracle/rman_bkup_merge.cmd
```

```
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.287.113
7018311
```

```
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.272.
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
=======================

Key     File S Completion Time Ckp SCN    Ckp Time        Sparse
------- ---- - --------------- ---------- --------------- ------
19      1    A 17-MAY-23       3009819    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSTEM_FNO-1_0h1sd7ae
        Tag: ORACOPYBKUPONFSXN_LEVEL_0

20      3    A 17-MAY-23       3009826    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSAUX_FNO-3_0i1sd7at
        Tag: ORACOPYBKUPONFSXN_LEVEL_0

21      4    A 17-MAY-23       3009830    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-4_0j1sd7b4
        Tag: ORACOPYBKUPONFSXN_LEVEL_0

27      5    A 17-MAY-23       2383520    12-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSTEM_FNO-5_0p1sd7cf
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 2, PDB Name: PDB$SEED

26      6    A 17-MAY-23       2383520    12-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSAUX_FNO-6_0o1sd7c8
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 2, PDB Name: PDB$SEED

34      7    A 17-MAY-23       3009907    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
7_101sd7dl
        Tag: ORACOPYBKUPONFSXN_LEVEL_0

33      8    A 17-MAY-23       2383520    12-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-8_0v1sd7di
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
```

```
        Container ID: 2, PDB Name: PDB$SEED

28      9    A 17-MAY-23       3009871    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSTEM_FNO-9_0q1sd7cm
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

22      10   A 17-MAY-23       3009849    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSAUX_FNO-10_0k1sd7bb
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

25      11   A 17-MAY-23       3009862    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-11_0n1sd7c1
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

35      12   A 17-MAY-23       3009909    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
12_111sd7dm
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

29      13   A 17-MAY-23       3009876    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSTEM_FNO-13_0r1sd7ct
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 4, PDB Name: DB1_PDB2

23      14   A 17-MAY-23       3009854    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSAUX_FNO-14_0l1sd7bi
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 4, PDB Name: DB1_PDB2

31      15   A 17-MAY-23       3009900    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-15_0t1sd7db
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 4, PDB Name: DB1_PDB2

36      16   A 17-MAY-23       3009911    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
```

```
16_121sd7dn
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 4, PDB Name: DB1_PDB2

30      17   A 17-MAY-23       3009895    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSTEM_FNO-17_0s1sd7d4
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 5, PDB Name: DB1_PDB3

24      18   A 17-MAY-23       3009858    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSAUX_FNO-18_0m1sd7bq
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 5, PDB Name: DB1_PDB3

32      19   A 17-MAY-23       3009903    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-19_0u1sd7de
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 5, PDB Name: DB1_PDB3

37      20   A 17-MAY-23       3009914    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
20_131sd7do
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 5, PDB Name: DB1_PDB3

4       21   A 17-MAY-23       3009019    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
21_021sd6pv
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

5       22   A 17-MAY-23       3009419    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
22_031sd6r2
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

6       23   A 17-MAY-23       3009460    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
23_041sd6s5
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1
```

```
7       24   A 17-MAY-23       3009473    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
24_051sd6t9
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

8       25   A 17-MAY-23       3009502    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
25_061sd6uc
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

9       26   A 17-MAY-23       3009548    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
26_071sd6vf
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

10      27   A 17-MAY-23       3009576    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
27_081sd70i
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

11      28   A 17-MAY-23       3009590    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
28_091sd71l
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

12      29   A 17-MAY-23       3009619    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
29_0a1sd72o
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

13      30   A 17-MAY-23       3009648    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
30_0b1sd73r
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

14      31   A 17-MAY-23       3009671    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
31_0c1sd74u
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
```

```
        Container ID: 3, PDB Name: DB1_PDB1

15      32   A 17-MAY-23       3009729    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
32_0d1sd762
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

16      33   A 17-MAY-23       3009743    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
33_0e1sd775
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

17      34   A 17-MAY-23       3009771    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
34_0f1sd788
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

18      35   A 17-MAY-23       3009805    17-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
35_0g1sd79b
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1


RMAN>
```

8. Report schema from Oracle RMAN command prompt to observe that current active database data files are in primary storage ASM +DATA disk group.

```
RMAN> report schema;

Report of database schema for database with db_unique_name DB1

List of Permanent Datafiles
===========================
File Size(MB) Tablespace           RB segs Datafile Name
---- -------- -------------------- ------- ------------------------
1    1060     SYSTEM               YES
+DATA/DB1/DATAFILE/system.257.1136666315
3    810      SYSAUX               NO
+DATA/DB1/DATAFILE/sysaux.258.1136666361
4    675      UNDOTBS1             YES
+DATA/DB1/DATAFILE/undotbs1.259.1136666385
```

```
5    400       PDB$SEED:SYSTEM       NO
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/system.266.11366
67165
6    460       PDB$SEED:SYSAUX       NO
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/sysaux.267.11366
67165
7    5         USERS                 NO
+DATA/DB1/DATAFILE/users.260.1136666387
8    230       PDB$SEED:UNDOTBS1     NO
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/undotbs1.268.113
6667165
9    400       DB1_PDB1:SYSTEM       YES
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/system.272.11366
68041
10   490       DB1_PDB1:SYSAUX       NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/sysaux.273.11366
68041
11   465       DB1_PDB1:UNDOTBS1     YES
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/undotbs1.271.113
6668041
12   5         DB1_PDB1:USERS        NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/users.275.113666
8057
13   400       DB1_PDB2:SYSTEM       YES
+DATA/DB1/FB867EA89ECF81C0E053630F1EACB901/DATAFILE/system.277.11366
68057
14   470       DB1_PDB2:SYSAUX       NO
+DATA/DB1/FB867EA89ECF81C0E053630F1EACB901/DATAFILE/sysaux.278.11366
68057
15   235       DB1_PDB2:UNDOTBS1     YES
+DATA/DB1/FB867EA89ECF81C0E053630F1EACB901/DATAFILE/undotbs1.276.113
6668057
16   5         DB1_PDB2:USERS        NO
+DATA/DB1/FB867EA89ECF81C0E053630F1EACB901/DATAFILE/users.280.113666
8071
17   400       DB1_PDB3:SYSTEM       YES
+DATA/DB1/FB867F8A4D4F821CE053630F1EAC69CC/DATAFILE/system.282.11366
68073
18   470       DB1_PDB3:SYSAUX       NO
+DATA/DB1/FB867F8A4D4F821CE053630F1EAC69CC/DATAFILE/sysaux.283.11366
68073
19   235       DB1_PDB3:UNDOTBS1     YES
+DATA/DB1/FB867F8A4D4F821CE053630F1EAC69CC/DATAFILE/undotbs1.281.113
6668073
20   5         DB1_PDB3:USERS        NO
+DATA/DB1/FB867F8A4D4F821CE053630F1EAC69CC/DATAFILE/users.285.113666
```

```
8087
21   4096     DB1_PDB1:SOE          NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.286.11370182
39
22   4096     DB1_PDB1:SOE          NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.287.11370183
11
23   4096     DB1_PDB1:SOE          NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.288.11370183
59
24   4096     DB1_PDB1:SOE          NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.289.11370184
05
25   4096     DB1_PDB1:SOE          NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.290.11370184
43
26   4096     DB1_PDB1:SOE          NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.291.11370184
81
27   4096     DB1_PDB1:SOE          NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.292.11370185
23
28   4096     DB1_PDB1:SOE          NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.293.11370187
07
29   4096     DB1_PDB1:SOE          NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.294.11370187
45
30   4096     DB1_PDB1:SOE          NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.295.11370187
87
31   4096     DB1_PDB1:SOE          NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.296.11370188
37
32   4096     DB1_PDB1:SOE          NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.297.11370189
35
33   4096     DB1_PDB1:SOE          NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.298.11370190
77
34   4096     DB1_PDB1:SOE          NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.299.11370191
17
35   4096     DB1_PDB1:SOE          NO
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/DATAFILE/soe.300.11370191
81
```

```
List of Temporary Files
=======================
File Size(MB) Tablespace           Maxsize(MB) Tempfile Name
---- -------- -------------------- ----------- --------------------
1    123      TEMP                 32767
+DATA/DB1/TEMPFILE/temp.265.1136666447
2    123      PDB$SEED:TEMP        32767
+DATA/DB1/FB864A929AEB79B9E053630F1EAC7046/TEMPFILE/temp.269.1136667
185
3    10240    DB1_PDB1:TEMP        32767
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/TEMPFILE/temp.274.1136668
051
4    123      DB1_PDB2:TEMP        32767
+DATA/DB1/FB867EA89ECF81C0E053630F1EACB901/TEMPFILE/temp.279.1136668
067
5    123      DB1_PDB3:TEMP        32767
+DATA/DB1/FB867F8A4D4F821CE053630F1EAC69CC/TEMPFILE/temp.284.1136668
081

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_031sd6r2
-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_0b1sd73r
-rw-r----- 1 oracle asm 4294975488 May 17 18:15 data_D-DB1_I-
1730530050_TS-SOE_FNO-31_0c1sd74u
-rw-r----- 1 oracle asm 4294975488 May 17 18:16 data_D-DB1_I-
1730530050_TS-SOE_FNO-32_0d1sd762
-rw-r----- 1 oracle asm 4294975488 May 17 18:16 data_D-DB1_I-
1730530050_TS-SOE_FNO-33_0e1sd775
-rw-r----- 1 oracle asm 4294975488 May 17 18:17 data_D-DB1_I-
1730530050_TS-SOE_FNO-34_0f1sd788
-rw-r----- 1 oracle asm 4294975488 May 17 18:17 data_D-DB1_I-
1730530050_TS-SOE_FNO-35_0g1sd79b
-rw-r----- 1 oracle asm  513810432 May 17 18:18 data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-10_0k1sd7bb
-rw-r----- 1 oracle asm  492838912 May 17 18:18 data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-14_0l1sd7bi
-rw-r----- 1 oracle asm  492838912 May 17 18:18 data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-18_0m1sd7bq
-rw-r----- 1 oracle asm  849354752 May 17 18:18 data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-3_0i1sd7at
-rw-r----- 1 oracle asm  482353152 May 17 18:18 data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-6_0o1sd7c8
-rw-r----- 1 oracle asm 1111498752 May 17 18:18 data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-1_0h1sd7ae
-rw-r----- 1 oracle asm  419438592 May 17 18:19 data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-13_0r1sd7ct
-rw-r----- 1 oracle asm  419438592 May 17 18:19 data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-17_0s1sd7d4
-rw-r----- 1 oracle asm  419438592 May 17 18:19 data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-5_0p1sd7cf
-rw-r----- 1 oracle asm  419438592 May 17 18:19 data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-9_0q1sd7cm
-rw-r----- 1 oracle asm  487596032 May 17 18:18 data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-11_0n1sd7c1
-rw-r----- 1 oracle asm  246423552 May 17 18:19 data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-15_0t1sd7db
-rw-r----- 1 oracle asm  246423552 May 17 18:19 data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-19_0u1sd7de
-rw-r----- 1 oracle asm  707796992 May 17 18:18 data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-4_0j1sd7b4
-rw-r----- 1 oracle asm  241180672 May 17 18:19 data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-8_0v1sd7di
-rw-r----- 1 oracle asm    5251072 May 17 18:19 data_D-DB1_I-
1730530050_TS-USERS_FNO-12_111sd7dm
-rw-r----- 1 oracle asm    5251072 May 17 18:19 data_D-DB1_I-
1730530050_TS-USERS_FNO-16_121sd7dn
-rw-r----- 1 oracle asm    5251072 May 17 18:19 data_D-DB1_I-
```

```
1730530050_TS-USERS_FNO-20_131sd7do
-rw-r----- 1 oracle asm    5251072 May 17 18:19 data_D-DB1_I-
1730530050_TS-USERS_FNO-7_101sd7dl
```

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 ~]$ sudo su
[root@ip-172-30-15-99 ec2-user]# su - oracle
Last login: Thu May 18 14:22:34 UTC 2023
[oracle@ip-172-30-15-99 ~]$ sqlplus / as 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;

        ID
----------
DT
---------------------------------------------------------------------------
-------
EVENT
---------------------------------------------------------------------------
------------
         1
18-MAY-23 02.35.37.000000 PM
test oracle incremental merge switch to copy


SQL>
```

2. Simulate a failure by shutdown abort database, then start up oracle in mount stage.

```
SQL> shutdown abort;
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.

Total System Global Area 1.2885E+10 bytes
Fixed Size                  9177880 bytes
Variable Size            1778384896 bytes
Database Buffers         1.1073E+10 bytes
Redo Buffers               24375296 bytes
Database mounted.
SQL>
```

3. As oracle user, connect to Oracle database via RMAN to switch database to copy.

```
RMAN> switch database to copy;

datafile 1 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-1_0h1sd7ae"
datafile 3 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-3_0i1sd7at"
```

```
datafile 4 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-4_0j1sd7b4"
datafile 5 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-5_0p1sd7cf"
datafile 6 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-6_0o1sd7c8"
datafile 7 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-7_101sd7dl"
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_091sd71l"
datafile 29 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-29_0a1sd72o"
datafile 30 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-30_0b1sd73r"
datafile 31 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-31_0c1sd74u"
datafile 32 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-32_0d1sd762"
datafile 33 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-33_0e1sd775"
datafile 34 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-34_0f1sd788"
datafile 35 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-35_0g1sd79b"
```

4. Recover and open database to bring it up to current from last incremental backup.

```
RMAN> recover database;

Starting recover at 18-MAY-23
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=392 device type=DISK
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup
set
destination for restore of datafile 00009: /nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SYSTEM_FNO-9_0q1sd7cm
destination for restore of datafile 00023: /nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-23_041sd6s5
destination for restore of datafile 00027: /nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-27_081sd70i
destination for restore of datafile 00031: /nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-31_0c1sd74u
destination for restore of datafile 00034: /nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SOE_FNO-34_0f1sd788
channel ORA_DISK_1: reading from backup piece
/nfsfsxn/oracopy/321sfous_98_1_1
channel ORA_DISK_1: piece handle=/nfsfsxn/oracopy/321sfous_98_1_1
tag=ORACOPYBKUPONFSXN_LEVEL_0
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
```

```
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup
set
destination for restore of datafile 00010: /nfsfsxn/oracopy/data_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/3i1sfov0_114_1_1
channel ORA_DISK_1: piece handle=/nfsfsxn/oracopy/3i1sfov0_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/3j1sfov0_115_1_1
channel ORA_DISK_1: piece handle=/nfsfsxn/oracopy/3j1sfov0_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;

NAME
--------------------------------------------------------------------
------------
+DATA/DB1/CONTROLFILE/current.261.1136666435
+DATA/DB1/FB864A929AEB79B9E053630F1EAC7046/TEMPFILE/temp.269.1136667
185
+DATA/DB1/FB867DA8C68C816EE053630F1EAC2BCF/TEMPFILE/temp.274.1136668
051
+DATA/DB1/FB867EA89ECF81C0E053630F1EACB901/TEMPFILE/temp.279.1136668
067
+DATA/DB1/FB867F8A4D4F821CE053630F1EAC69CC/TEMPFILE/temp.284.1136668
081
+DATA/DB1/ONLINELOG/group_1.262.1136666437
+DATA/DB1/ONLINELOG/group_2.263.1136666437
+DATA/DB1/ONLINELOG/group_3.264.1136666437
+DATA/DB1/TEMPFILE/temp.265.1136666447
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-21_021sd6pv
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-22_031sd6r2

NAME
--------------------------------------------------------------------
------------
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-23_041sd6s5
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-24_051sd6t9
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-25_061sd6uc
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-26_071sd6vf
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-27_081sd70i
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-28_091sd71l
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-29_0a1sd72o
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-30_0b1sd73r
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-31_0c1sd74u
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-32_0d1sd762
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-33_0e1sd775
```

```
NAME
--------------------------------------------------------------------
------------
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-34_0f1sd788
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-35_0g1sd79b
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-10_0k1sd7bb
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-14_0l1sd7bi
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-18_0m1sd7bq
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-3_0i1sd7at
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-6_0o1sd7c8
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-13_0r1sd7ct
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-17_0s1sd7d4
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-1_0h1sd7ae
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-5_0p1sd7cf

NAME
--------------------------------------------------------------------
------------
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-9_0q1sd7cm
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-11_0n1sd7c1
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-15_0t1sd7db
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-19_0u1sd7de
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-4_0j1sd7b4
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-8_0v1sd7di
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-12_111sd7dm
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-16_121sd7dn
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-20_131sd7do
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-7_101sd7dl

43 rows selected.

SQL>
```

6. From SQL plus, check the content of test table we have inserted before the switch over to copy

```
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> select * from test;

        ID
----------
DT
--------------------------------------------------------------------
-------
EVENT
--------------------------------------------------------------------
------------
         1
18-MAY-23 02.35.37.000000 PM
test oracle incremental merge switch to copy


SQL>
```

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;
```

```
        ID
----------
DT
---------------------------------------------------------------------
-------
EVENT
---------------------------------------------------------------------
------------
         1
18-MAY-23 02.35.37.000000 PM
test oracle incremental merge switch to copy

         2
30-MAY-23 05.23.11.000000 PM
test recovery on a new EC2 instance host with image copy on FSxN


SQL>
```

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. Privison a new EC2 DB instance host ora_02 with same OS and version via AWS EC2 console. Configure OS kernal with same patches as primary EC2 DB server host, Oracle preinstall RPM, and add swap space to the host as well. Install same version and patches of Oracle as in primary EC2 DB server host with software only option. These tasks can be automated with NetApp automation toolkit as available from below links.

   Toolkit: na_oracle19c_deploy
   Documentation: Automated Deployment of Oracle19c for ONTAP on NFS

5. Configure oracle environment similiarly 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 avilability, 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 provsioned 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,noin
   tr
   ```

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 6i1tf6b9_210_1_1
   -rw-r-----. 1 oracle 54331     475136 May 30 15:29 6j1tf6bb_211_1_1
   -rw-r-----. 1 oracle 54331   48660480 May 30 15:29 6g1tf6b9_208_1_1
   -rw-r-----. 1 oracle 54331     589824 May 30 15:29 6l1tf6bb_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
   ```

```
-rw-r-----. 1 oracle 54331 4294975488 May 30 17:26 data_D-DB1_I-
1730530050_TS-SOE_FNO-23_3q1t4ut3
-rw-r-----. 1 oracle 54331 4294975488 May 30 17:26 data_D-DB1_I-
1730530050_TS-SOE_FNO-21_3o1t4ut2
-rw-r-----. 1 oracle 54331 4294975488 May 30 17:26 data_D-DB1_I-
1730530050_TS-SOE_FNO-27_461t4vt7
-rw-r-----. 1 oracle 54331 4294975488 May 30 17:26 data_D-DB1_I-
1730530050_TS-SOE_FNO-25_3s1t4v1a
-rw-r-----. 1 oracle 54331 4294975488 May 30 17:26 data_D-DB1_I-
1730530050_TS-SOE_FNO-22_3p1t4ut3
-rw-r-----. 1 oracle 54331 4294975488 May 30 17:26 data_D-DB1_I-
1730530050_TS-SOE_FNO-31_4a1t5015
-rw-r-----. 1 oracle 54331 4294975488 May 30 17:26 data_D-DB1_I-
1730530050_TS-SOE_FNO-29_481t4vt7
-rw-r-----. 1 oracle 54331 4294975488 May 30 17:26 data_D-DB1_I-
1730530050_TS-SOE_FNO-34_4d1t5058
-rw-r-----. 1 oracle 54331 4294975488 May 30 17:26 data_D-DB1_I-
1730530050_TS-SOE_FNO-26_451t4vt7
-rw-r-----. 1 oracle 54331 4294975488 May 30 17:26 data_D-DB1_I-
1730530050_TS-SOE_FNO-24_3r1t4ut3
-rw-r-----. 1 oracle 54331  555753472 May 30 17:26 data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-10_4i1t5083
-rw-r-----. 1 oracle 54331  429924352 May 30 17:26 data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-9_4n1t509m
-rw-r-----. 1 oracle 54331 4294975488 May 30 17:26 data_D-DB1_I-
1730530050_TS-SOE_FNO-30_491t5014
-rw-r-----. 1 oracle 54331 4294975488 May 30 17:26 data_D-DB1_I-
1730530050_TS-SOE_FNO-28_471t4vt7
-rw-r-----. 1 oracle 54331 4294975488 May 30 17:26 data_D-DB1_I-
1730530050_TS-SOE_FNO-35_4e1t5059
-rw-r-----. 1 oracle 54331 4294975488 May 30 17:26 data_D-DB1_I-
1730530050_TS-SOE_FNO-32_4b1t501u
-rw-r-----. 1 oracle 54331  487596032 May 30 17:26 data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-11_4l1t508t
-rw-r-----. 1 oracle 54331 4294975488 May 30 17:26 data_D-DB1_I-
1730530050_TS-SOE_FNO-33_4c1t501v
-rw-r-----. 1 oracle 54331    5251072 May 30 17:26 data_D-DB1_I-
1730530050_TS-USERS_FNO-12_4v1t50aa
-rw-r-----. 1 oracle 54331 1121984512 May 30 17:26 data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-1_4f1t506m
-rw-r-----. 1 oracle 54331  707796992 May 30 17:26 data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-4_4h1t5083
-rw-r-----. 1 oracle 54331  534781952 May 30 17:26 data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-14_4j1t508s
-rw-r-----. 1 oracle 54331  429924352 May 30 17:26 data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-13_4o1t509m
```

```
-rw-r-----. 1 oracle 54331  429924352 May 30 17:26 data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-17_4p1t509m
-rw-r-----. 1 oracle 54331  534781952 May 30 17:26 data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-18_4k1t508t
-rw-r-----. 1 oracle 54331 1027612672 May 30 17:26 data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-3_4g1t506m
-rw-r-----. 1 oracle 54331    5251072 May 30 17:26 data_D-DB1_I-
1730530050_TS-USERS_FNO-7_4u1t50a6
-rw-r-----. 1 oracle 54331  246423552 May 30 17:26 data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-15_4r1t50a6
-rw-r-----. 1 oracle 54331    5251072 May 30 17:26 data_D-DB1_I-
1730530050_TS-USERS_FNO-16_501t50ad
-rw-r-----. 1 oracle 54331  246423552 May 30 17:26 data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-19_4s1t50a6
-rw-r-----. 1 oracle 54331    5251072 May 30 17:26 data_D-DB1_I-
1730530050_TS-USERS_FNO-20_511t50ad
-rw-r-----. 1 oracle 54331 2318712832 May 30 17:32 721tfd6b_226_1_1
-rw-r-----. 1 oracle 54331 1813143552 May 30 17:33 701tfd6a_224_1_1
-rw-r-----. 1 oracle 54331     966656 May 30 17:33 731tfdic_227_1_1
-rw-r-----. 1 oracle 54331    5980160 May 30 17:33 751tfdij_229_1_1
-rw-r-----. 1 oracle 54331     458752 May 30 17:33 761tfdin_230_1_1
-rw-r-----. 1 oracle 54331     458752 May 30 17:33 771tfdiq_231_1_1
-rw-r-----. 1 oracle 54331   11091968 May 30 17:33 741tfdij_228_1_1
-rw-r-----. 1 oracle 54331     401408 May 30 17:33 791tfdit_233_1_1
-rw-r-----. 1 oracle 54331 2070708224 May 30 17:33 6v1tfd6a_223_1_1
-rw-r-----. 1 oracle 54331     376832 May 30 17:33 7a1tfdit_234_1_1
-rw-r-----. 1 oracle 54331 1874903040 May 30 17:33 711tfd6b_225_1_1
-rw-r-----. 1 oracle 54331     303104 May 30 17:33 7c1tfdiu_236_1_1
-rw-r-----. 1 oracle 54331     319488 May 30 17:33 7d1tfdiv_237_1_1
-rw-r-----. 1 oracle 54331      57344 May 30 17:33 7f1tfdiv_239_1_1
-rw-r-----. 1 oracle 54331      57344 May 30 17:33 7g1tfdiv_240_1_1
-rw-r-----. 1 oracle 54331      57344 May 30 17:33 7h1tfdiv_241_1_1
-rw-r--r--. 1 oracle 54331      12720 May 30 17:33 db1_ctl.sql
-rw-r-----. 1 oracle 54331   11600384 May 30 17:54 bct_db1.ctf
```

8. Verify the available Oracle archived logs on the FSxN NFS mount for recovery and note the last log file log sequency number. In this case, it is 175. Our recovery point is up to log sequency number 176.

```
 [ec2-user@ip-172-30-15-124 ~]$ ls -ltr
/nfsfsxn/archlog/DB1/archivelog/2023_05_30
total 5714400
-r--r-----. 1 oracle 54331    321024 May 30 14:59
o1_mf_1_140__003t9mvn_.arc
-r--r-----. 1 oracle 54331  48996352 May 30 15:29
o1_mf_1_141__01t9qf6r_.arc
-r--r-----. 1 oracle 54331 167477248 May 30 15:44
```

```
o1_mf_1_142__02n3x2qb_.arc
-r--r-----. 1 oracle 54331 165684736 May 30 15:46
o1_mf_1_143__02rotwyb_.arc
-r--r-----. 1 oracle 54331 165636608 May 30 15:49
o1_mf_1_144__02x563wh_.arc
-r--r-----. 1 oracle 54331 168408064 May 30 15:51
o1_mf_1_145__031kg2co_.arc
-r--r-----. 1 oracle 54331 169446400 May 30 15:54
o1_mf_1_146__035xpcdt_.arc
-r--r-----. 1 oracle 54331 167595520 May 30 15:56
o1_mf_1_147__03bds8qf_.arc
-r--r-----. 1 oracle 54331 169270272 May 30 15:59
o1_mf_1_148__03gyt7rx_.arc
-r--r-----. 1 oracle 54331 170712576 May 30 16:01
o1_mf_1_149__03mfxl7v_.arc
-r--r-----. 1 oracle 54331 170744832 May 30 16:04
o1_mf_1_150__03qzz0ty_.arc
-r--r-----. 1 oracle 54331 169380864 May 30 16:06
o1_mf_1_151__03wgxdry_.arc
-r--r-----. 1 oracle 54331 169833984 May 30 16:09
o1_mf_1_152__040y85v3_.arc
-r--r-----. 1 oracle 54331 165134336 May 30 16:20
o1_mf_1_153__04ox946w_.arc
-r--r-----. 1 oracle 54331 169929216 May 30 16:22
o1_mf_1_154__04rbv7n8_.arc
-r--r-----. 1 oracle 54331 171903488 May 30 16:23
o1_mf_1_155__04tv1yvn_.arc
-r--r-----. 1 oracle 54331 179061248 May 30 16:25
o1_mf_1_156__04xgfjtl_.arc
-r--r-----. 1 oracle 54331 173593088 May 30 16:26
o1_mf_1_157__04zyg8hw_.arc
-r--r-----. 1 oracle 54331 175999488 May 30 16:27
o1_mf_1_158__052gp9mt_.arc
-r--r-----. 1 oracle 54331 179092992 May 30 16:29
o1_mf_1_159__0551wk7s_.arc
-r--r-----. 1 oracle 54331 175524352 May 30 16:30
o1_mf_1_160__057l46my_.arc
-r--r-----. 1 oracle 54331 173949440 May 30 16:32
o1_mf_1_161__05b2dmwp_.arc
-r--r-----. 1 oracle 54331 184166912 May 30 16:33
o1_mf_1_162__05drbj8n_.arc
-r--r-----. 1 oracle 54331 173026816 May 30 16:35
o1_mf_1_163__05h8lm1h_.arc
-r--r-----. 1 oracle 54331 174286336 May 30 16:36
o1_mf_1_164__05krsqmh_.arc
-r--r-----. 1 oracle 54331 166092288 May 30 16:37
```

```
o1_mf_1_165__05n378pw_.arc
-r--r-----. 1 oracle 54331 177640960 May 30 16:39
o1_mf_1_166__05pmg74l_.arc
-r--r-----. 1 oracle 54331 173972992 May 30 16:40
o1_mf_1_167__05s3o01r_.arc
-r--r-----. 1 oracle 54331 178474496 May 30 16:41
o1_mf_1_168__05vmwt34_.arc
-r--r-----. 1 oracle 54331 177694208 May 30 16:43
o1_mf_1_169__05y45qdd_.arc
-r--r-----. 1 oracle 54331 170814976 May 30 16:44
o1_mf_1_170__060kgh33_.arc
-r--r-----. 1 oracle 54331 177325056 May 30 16:46
o1_mf_1_171__0631tvgv_.arc
-r--r-----. 1 oracle 54331 164455424 May 30 16:47
o1_mf_1_172__065d94fq_.arc
-r--r-----. 1 oracle 54331 178252288 May 30 16:48
o1_mf_1_173__067wnwy8_.arc
-r--r-----. 1 oracle 54331 170579456 May 30 16:50
o1_mf_1_174__06b9zdh8_.arc
-r--r-----. 1 oracle 54331  93928960 May 30 17:26
o1_mf_1_175__08c7jc2b_.arc
[ec2-user@ip-172-30-15-124 ~]$
```

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 FSxN NFS 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
=======================

Key     File S Completion Time Ckp SCN    Ckp Time        Sparse
------- ---- - --------------- ---------- --------------- ------
316     1    A 30-MAY-23       4120170    30-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSTEM_FNO-1_4f1t506m
        Tag: ORACOPYBKUPONFSXN_LEVEL_0

322     3    A 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     4    A 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     5    A 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     6    A 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     7    A 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     8    A 26-MAY-23       2383520    12-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-8_4t1t50a6
```

```
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 2, PDB Name: PDB$SEED

308     9    A 30-MAY-23       4120158    30-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSTEM_FNO-9_4n1t509m
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

307     10   A 30-MAY-23       4120166    30-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSAUX_FNO-10_4i1t5083
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

313     11   A 30-MAY-23       4120154    30-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-11_4l1t508t
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

315     12   A 30-MAY-23       4120162    30-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
12_4v1t50aa
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

319     13   A 30-MAY-23       4120191    30-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSTEM_FNO-13_4o1t509m
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 4, PDB Name: DB1_PDB2

318     14   A 30-MAY-23       4120183    30-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSAUX_FNO-14_4j1t508s
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 4, PDB Name: DB1_PDB2

324     15   A 30-MAY-23       4120199    30-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-15_4r1t50a6
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 4, PDB Name: DB1_PDB2

325     16   A 30-MAY-23       4120211    30-MAY-23       NO
```

```
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
16_501t50ad
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 4, PDB Name: DB1_PDB2

320     17   A 30-MAY-23       4120195    30-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSTEM_FNO-17_4p1t509m
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 5, PDB Name: DB1_PDB3

321     18   A 30-MAY-23       4120187    30-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSAUX_FNO-18_4k1t508t
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 5, PDB Name: DB1_PDB3

326     19   A 30-MAY-23       4120203    30-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-19_4s1t50a6
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 5, PDB Name: DB1_PDB3

327     20   A 30-MAY-23       4120216    30-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
20_511t50ad
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 5, PDB Name: DB1_PDB3

298     21   A 30-MAY-23       4120166    30-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
21_3o1t4ut2
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

302     22   A 30-MAY-23       4120154    30-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
22_3p1t4ut3
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

297     23   A 30-MAY-23       4120158    30-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
23_3q1t4ut3
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1
```

```
306     24   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

300     25   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

305     26   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

299     27   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

310     28   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

303     29   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

309     30   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

301     31   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
```

```
=======================
File Name:
/nfsfsxn/archlog/DB1/autobackup/2023_05_30/o1_mf_s_1138210401__08qlx
rrr_.bkp

datafile 1 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-1_4f1t506m"
datafile 3 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-3_4g1t506m"
datafile 4 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-4_4h1t5083"
datafile 5 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-5_4q1t509n"
datafile 6 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-6_4m1t508t"
datafile 7 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-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_4v1t50aa"
datafile 13 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SYSTEM_FNO-13_4o1t509m"
datafile 14 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SYSAUX_FNO-14_4j1t508s"
datafile 15 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-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-
```

```
DB1_I-1730530050_TS-SOE_FNO-22_3p1t4ut3"
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
```

```
c
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
c
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
c
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
c
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
c
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
c
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
c
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
c
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
c
archived log for thread 1 with sequence 151 is already on disk as
file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_151__03wgxdry_.ar
c
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
c
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
c
archived log for thread 1 with sequence 154 is already on disk as
```

```
file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_154__04rbv7n8_.ar
c
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
c
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
c
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
c
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
c
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
c
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__057l46my_.ar
c
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
c
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
c
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__05h8lm1h_.ar
c
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
c
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
```

```
c
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__05pmg74l_.ar
c
archived log for thread 1 with sequence 167 is already on disk as
file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_167__05s3o01r_.ar
c
archived log for thread 1 with sequence 168 is already on disk as
file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_168__05vmwt34_.ar
c
archived log for thread 1 with sequence 169 is already on disk as
file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_169__05y45qdd_.ar
c
archived log for thread 1 with sequence 170 is already on disk as
file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_170__060kgh33_.ar
c
archived log for thread 1 with sequence 171 is already on disk as
file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_171__0631tvgv_.ar
c
archived log for thread 1 with sequence 172 is already on disk as
file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_172__065d94fq_.ar
c
archived log for thread 1 with sequence 173 is already on disk as
file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_173__067wnwy8_.ar
c
archived log for thread 1 with sequence 174 is already on disk as
file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_174__06b9zdh8_.ar
c
archived log for thread 1 with sequence 175 is already on disk as
file
/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_175__08c7jc2b_.ar
c
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_142__02n3x2q
b_.arc thread=1 sequence=142
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_143__02rotwy
```

```
b_.arc thread=1 sequence=143
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_144__02x563w
h_.arc thread=1 sequence=144
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_145__031kg2c
o_.arc thread=1 sequence=145
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_146__035xpcd
t_.arc thread=1 sequence=146
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_147__03bds8q
f_.arc thread=1 sequence=147
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_148__03gyt7r
x_.arc thread=1 sequence=148
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_149__03mfxl7
v_.arc thread=1 sequence=149
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_150__03qzz0t
y_.arc thread=1 sequence=150
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_151__03wgxdr
y_.arc thread=1 sequence=151
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_152__040y85v
3_.arc thread=1 sequence=152
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_153__04ox946
w_.arc thread=1 sequence=153
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_154__04rbv7n
8_.arc thread=1 sequence=154
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_155__04tv1yv
n_.arc thread=1 sequence=155
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_156__04xgfjt
l_.arc thread=1 sequence=156
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_157__04zyg8h
w_.arc thread=1 sequence=157
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_158__052gp9m
t_.arc thread=1 sequence=158
```

```
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_159__0551wk7
s_.arc thread=1 sequence=159
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_160__057l46m
y_.arc thread=1 sequence=160
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_161__05b2dmw
p_.arc thread=1 sequence=161
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_162__05drbj8
n_.arc thread=1 sequence=162
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_163__05h8lm1
h_.arc thread=1 sequence=163
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_164__05krsqm
h_.arc thread=1 sequence=164
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_165__05n378p
w_.arc thread=1 sequence=165
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_166__05pmg74
l_.arc thread=1 sequence=166
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_167__05s3o01
r_.arc thread=1 sequence=167
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_168__05vmwt3
4_.arc thread=1 sequence=168
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_169__05y45qd
d_.arc thread=1 sequence=169
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_170__060kgh3
3_.arc thread=1 sequence=170
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_171__0631tvg
v_.arc thread=1 sequence=171
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_172__065d94f
q_.arc thread=1 sequence=172
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_173__067wnwy
8_.arc thread=1 sequence=173
archived log file
```

```
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_174__06b9zdh
8_.arc thread=1 sequence=174
archived log file
name=/nfsfsxn/archlog/DB1/archivelog/2023_05_30/o1_mf_1_175__08c7jc2
b_.arc thread=1 sequence=175
media recovery complete, elapsed time: 00:48:34
Finished recover at 31-MAY-23
```

For faster recovery, enable parallel sessions with recovery_parallelism parameter or specify degree of parallel in recovery command for database recovery: `RECOVER DATABASE PARALLEL (DEGREE d INSTANCES DEFAULT);`. In general, degrees of parallelism should be equal to number of CPU cores on the host.

18. Exit RMAN, login to Oracle as oracle user via sqlplus to open database and reset log after an incomplete recovery.

```
SQL> select name, open_mode from v$database;

NAME      OPEN_MODE
--------- --------------------
DB1       MOUNTED

SQL> select member from v$logfile;

MEMBER
--------------------------------------------------------------------
------------
+DATA/DB1/ONLINELOG/group_3.264.1136666437
+DATA/DB1/ONLINELOG/group_2.263.1136666437
+DATA/DB1/ONLINELOG/group_1.262.1136666437

SQL> alter database rename file
'+DATA/DB1/ONLINELOG/group_1.262.1136666437' to
'/nfsfsxn/oracopy/redo01.log';

Database altered.

SQL> alter database rename file
'+DATA/DB1/ONLINELOG/group_2.263.1136666437' to
'/nfsfsxn/oracopy/redo02.log';

Database altered.

SQL> alter database rename file
'+DATA/DB1/ONLINELOG/group_3.264.1136666437' to
'/nfsfsxn/oracopy/redo03.log';

Database altered.

SQL> alter database open resetlogs;

Database altered.
```

19. Validate the database restored to new host that has the row we have inserted before primary database failure.

```
SQL> show 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> select * from test;

        ID DT
EVENT
----------
--------------------------------------------------------------------
-------
--------------------------------------------------------------------
--------------------------------
         1 18-MAY-23 02.35.37.000000 PM
test oracle incremental merge switch to copy
         2 30-MAY-23 05.23.11.000000 PM
test recovery on a new EC2 instance host with image copy on FSxN
```

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,noin
tr  0       0

Update the Oracle init file from primary databse init file backup
that is restored to /tmp/archive and create spfile as needed.
```

This completes the Oracle VLDB database recovery from backup image copy on FSxN NFS file system to a new EC2 DB instance host.

### Clone Oracle standby image copy for other use cases

Another benefit of using AWS FSx ONTAP for staging Oracle VLDB image copy is that it can be FlexCloned to serve many other purposes with minimal additional storage investment. In the following use case, we demonstrate how to snapshot and clone the staging NFS volume on FSx ONTAP for other Oracle use cases such as DEV, UAT, etc.

1. We begin with inserting a row into the same test table we have created before.

```
 SQL> insert into test values (3, sysdate, 'test clone on a new EC2
instance host with image copy on FSxN');

1 row created.

SQL> select * from test;

        ID
----------
DT
--------------------------------------------------------------------
-------
EVENT
--------------------------------------------------------------------
------------
         1
18-MAY-23 02.35.37.000000 PM
test oracle incremental merge switch to copy

         2
30-MAY-23 05.23.11.000000 PM
test recovery on a new EC2 instance host with image copy on FSxN

        ID
----------
DT
--------------------------------------------------------------------
-------
EVENT
--------------------------------------------------------------------
------------


         3
05-JUN-23 03.19.46.000000 PM
test clone on a new EC2 instance host with image copy on FSxN


SQL>
```

2. Take a RMAN backup and merge to FSx ONTAP database image copy so that the transaction will be captured in the backup set on FSx NFS mount but not merged into copy until cloned database is recovered.

```
RMAN> @/home/oracle/rman_bkup_merge.cmd
```

3. Login to FSx cluster via ssh as fsxadmin user to observe the snapshots created by scheduled backup policy - oracle and take an one-off snapshot so that it will include the transaction we commited in step 1.

```
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%
--------- ------------ ------------ ---------- ---- ----------
---------- -----
svm_ora   db1_20230605of
                       aggr1        online     RW         200GB
116.6GB   38%

FsxId06c3c8b2a7bd56458::>
```

5. Turn off snapshot policy for the cloned volume as it inherits parent volume snapshot policy unless you want to protect the cloned volume, then leave it alone.

```
FsxId06c3c8b2a7bd56458::> vol modify -volume db1_20230605of
-snapshot-policy none

Warning: You are changing the Snapshot policy on volume
"db1_20230605of" to "none". Snapshot copies on this volume that do
not match any of the prefixes of the new Snapshot policy will not be
deleted. However, when the new Snapshot policy
         takes effect, depending on the new retention count, any
existing Snapshot copies that continue to use the same prefixes
might be deleted. See the 'volume modify' man page for more
information.
Do you want to continue? {y|n}: y
Volume modify successful on volume db1_20230605of of Vserver
svm_ora.

FsxId06c3c8b2a7bd56458::>
```

6. Login to a new EC2 Linux instance with Oracle software pre-installed with same version and patch level as your primary Oracle EC2 instance and mount the cloned volume.

```
[ec2-user@ip-172-30-15-124 ~]$ sudo mkdir /nfsfsxn
[ec2-user@ip-172-30-15-124 ~]$ sudo mount -t nfs
172.30.15.19:/db1_20230605of /nfsfsxn -o
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=262144,wsize=262144,noin
tr
```

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

```
[ec2-user@ip-172-30-15-124 ~]$ ls -ltr /nfsfsxn/oracopy
total 79450332
-rw-r----- 1 oracle 54331  482353152 Jun  1 19:02 data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-6_891tkrhr
-rw-r----- 1 oracle 54331  419438592 Jun  1 19:03 data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-5_8d1tkril
-rw-r----- 1 oracle 54331  241180672 Jun  1 19:03 data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-8_8g1tkrj7
-rw-r----- 1 oracle 54331  912506880 Jun  1 20:21 8n1tkvv2_279_1_1
-rw-r----- 1 oracle 54331     925696 Jun  1 20:21 8q1tl05i_282_1_1
-rw-r----- 1 oracle 54331 1169014784 Jun  1 20:21 8p1tkvv2_281_1_1
-rw-r----- 1 oracle 54331    6455296 Jun  1 20:21 8r1tl05m_283_1_1
-rw-r----- 1 oracle 54331     139264 Jun  1 20:21 8t1tl05t_285_1_1
-rw-r----- 1 oracle 54331    3514368 Jun  1 20:21 8s1tl05t_284_1_1
-rw-r----- 1 oracle 54331     139264 Jun  1 20:21 8u1tl060_286_1_1
-rw-r----- 1 oracle 54331     425984 Jun  1 20:21 901tl062_288_1_1
-rw-r----- 1 oracle 54331     344064 Jun  1 20:21 911tl062_289_1_1
-rw-r----- 1 oracle 54331     245760 Jun  1 20:21 931tl063_291_1_1
-rw-r----- 1 oracle 54331     237568 Jun  1 20:21 941tl064_292_1_1
-rw-r----- 1 oracle 54331      57344 Jun  1 20:21 961tl065_294_1_1
-rw-r----- 1 oracle 54331      57344 Jun  1 20:21 971tl066_295_1_1
-rw-r----- 1 oracle 54331      57344 Jun  1 20:21 981tl067_296_1_1
-rw-r----- 1 oracle 54331 1040760832 Jun  1 20:23 8m1tkvv2_278_1_1
-rw-r----- 1 oracle 54331  932847616 Jun  1 20:24 8o1tkvv2_280_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_8a1tkrhr
-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_871tkrhr
-rw-r----- 1 oracle 54331  534781952 Jun  5 15:21 data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-18_881tkrhr
```

```
-rw-r----- 1 oracle 54331  429924352 Jun  5 15:21 data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-13_8b1tkril
-rw-r----- 1 oracle 54331  429924352 Jun  5 15:21 data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-17_8c1tkril
-rw-r----- 1 oracle 54331  246423552 Jun  5 15:21 data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-15_8e1tkril
-rw-r----- 1 oracle 54331  246423552 Jun  5 15:21 data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-19_8f1tkrj4
-rw-r----- 1 oracle 54331    5251072 Jun  5 15:21 data_D-DB1_I-
1730530050_TS-USERS_FNO-7_8h1tkrj9
-rw-r----- 1 oracle 54331    5251072 Jun  5 15:21 data_D-DB1_I-
1730530050_TS-USERS_FNO-16_8j1tkrja
-rw-r----- 1 oracle 54331    5251072 Jun  5 15:21 data_D-DB1_I-
1730530050_TS-USERS_FNO-20_8k1tkrjb
-rw-r----- 1 oracle 54331    5251072 Jun  5 15:21 data_D-DB1_I-
1730530050_TS-USERS_FNO-12_8i1tkrj9
-rw-r----- 1 oracle 54331  555753472 Jun  5 15:21 data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-10_861tkrgo
-rw-r----- 1 oracle 54331  796925952 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-11_841tkrf2
-rw-r----- 1 oracle 54331 4294975488 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-SOE_FNO-21_7j1tkqk6
-rw-r----- 1 oracle 54331 4294975488 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-SOE_FNO-34_801tkram
-rw-r----- 1 oracle 54331 4294975488 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-SOE_FNO-29_7r1tkr32
-rw-r----- 1 oracle 54331 4294975488 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-SOE_FNO-25_7n1tkqrh
-rw-r----- 1 oracle 54331 4294975488 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-SOE_FNO-31_7t1tkr3i
-rw-r----- 1 oracle 54331 4294975488 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-SOE_FNO-33_7v1tkra6
-rw-r----- 1 oracle 54331 4294975488 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-SOE_FNO-23_7l1tkqk6
-rw-r----- 1 oracle 54331 4294975488 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-SOE_FNO-27_7p1tkqrq
-rw-r----- 1 oracle 54331 4294975488 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-SOE_FNO-35_811tkrap
-rw-r----- 1 oracle 54331 4294975488 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-SOE_FNO-32_7u1tkr42
-rw-r----- 1 oracle 54331 4294975488 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-SOE_FNO-22_7k1tkqk6
-rw-r----- 1 oracle 54331 4294975488 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-SOE_FNO-24_7m1tkqk6
-rw-r----- 1 oracle 54331 4294975488 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-SOE_FNO-28_7q1tkqs1
```

```
-rw-r----- 1 oracle 54331 4294975488 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-SOE_FNO-30_7s1tkr3a
-rw-r----- 1 oracle 54331 4294975488 Jun  5 15:22 data_D-DB1_I-
1730530050_TS-SOE_FNO-26_7o1tkqrj
-rw-r----- 1 oracle 54331 1241432064 Jun  5 15:30 9d1tv06n_301_1_1
-rw-r----- 1 oracle 54331 1019805696 Jun  5 15:31 9a1tv06m_298_1_1
-rw-r----- 1 oracle 54331    4612096 Jun  5 15:31 9e1tv0ld_302_1_1
-rw-r----- 1 oracle 54331  967163904 Jun  5 15:31 9b1tv06n_299_1_1
-rw-r----- 1 oracle 54331   31563776 Jun  5 15:31 9g1tv0lt_304_1_1
-rw-r----- 1 oracle 54331     319488 Jun  5 15:31 9h1tv0lt_305_1_1
-rw-r----- 1 oracle 54331     335872 Jun  5 15:31 9i1tv0m0_306_1_1
-rw-r----- 1 oracle 54331     565248 Jun  5 15:31 9k1tv0m1_308_1_1
-rw-r----- 1 oracle 54331     581632 Jun  5 15:31 9l1tv0m5_309_1_1
-rw-r----- 1 oracle 54331   54345728 Jun  5 15:31 9f1tv0lt_303_1_1
-rw-r----- 1 oracle 54331     368640 Jun  5 15:31 9n1tv0m5_311_1_1
-rw-r----- 1 oracle 54331     385024 Jun  5 15:31 9o1tv0m6_312_1_1
-rw-r----- 1 oracle 54331  985858048 Jun  5 15:31 9c1tv06n_300_1_1
-rw-r----- 1 oracle 54331      57344 Jun  5 15:31 9q1tv0m7_314_1_1
-rw-r----- 1 oracle 54331      57344 Jun  5 15:31 9r1tv0m8_315_1_1
-rw-r----- 1 oracle 54331      57344 Jun  5 15:31 9s1tv0m9_316_1_1
-rw-r--r-- 1 oracle 54331      12720 Jun  5 15:31 db1_ctl.sql
-rw-r----- 1 oracle 54331   11600384 Jun  5 15:48 bct_db1.ctf
[ec2-user@ip-172-30-15-124 ~]$

[oracle@ip-172-30-15-124 ~]$ ls -l
/nfsfsxn/archlog/DB1/archivelog/2023_06_05
total 2008864
-rw-r----- 1 oracle 54331    729088 Jun  5 14:38
o1_mf_1_190_l7vwvvt9_.arc
-rw-r----- 1 oracle 54331 166651904 Jun  5 14:44
o1_mf_1_191_l7vx6vmg_.arc
-rw-r----- 1 oracle 54331 167406080 Jun  5 14:47
o1_mf_1_192_l7vxctms_.arc
-rw-r----- 1 oracle 54331 166868992 Jun  5 14:49
o1_mf_1_193_l7vxjjps_.arc
-rw-r----- 1 oracle 54331 166087168 Jun  5 14:52
o1_mf_1_194_l7vxnxrh_.arc
-rw-r----- 1 oracle 54331 175210496 Jun  5 14:54
o1_mf_1_195_l7vxswv5_.arc
-rw-r----- 1 oracle 54331 167078400 Jun  5 14:57
o1_mf_1_196_l7vxylwp_.arc
-rw-r----- 1 oracle 54331 169701888 Jun  5 14:59
o1_mf_1_197_l7vy3cyw_.arc
-rw-r----- 1 oracle 54331 167845376 Jun  5 15:02
o1_mf_1_198_l7vy8245_.arc
-rw-r----- 1 oracle 54331 170763776 Jun  5 15:05
```

```
o1_mf_1_199_l7vydv4c_.arc
-rw-r----- 1 oracle 54331 193853440 Jun  5 15:07
o1_mf_1_200_l7vykf23_.arc
-rw-r----- 1 oracle 54331 165523968 Jun  5 15:09
o1_mf_1_201_l7vyp1dh_.arc
-rw-r----- 1 oracle 54331 161117184 Jun  5 15:12
o1_mf_1_202_l7vyvrm5_.arc
-rw-r----- 1 oracle 54331  10098176 Jun  5 15:21
o1_mf_1_203_l7vzdfwm_.arc
```

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, lanuch RMAN to run recovery. Following are command steps and output.

```
[oracle@ip-172-30-15-124 dbs]$ rman target / nocatalog

Recovery Manager: Release 19.0.0.0.0 - Production on Wed Jun 7
14:44:33 2023
Version 19.18.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights
reserved.

connected to target database (not started)

RMAN> startup nomount;

Oracle instance started

Total System Global Area   10737418000 bytes

Fixed Size                     9174800 bytes
Variable Size               1577058304 bytes
Database Buffers            9126805504 bytes
Redo Buffers                  24379392 bytes

RMAN> set dbid = 1730530050;

executing command: SET DBID

RMAN> restore controlfile from autobackup;

Starting restore at 07-JUN-23
allocated channel: ORA_DISK_1
```

```
channel ORA_DISK_1: SID=2 device type=DISK

recovery area destination: /nfsfsxn/archlog/
database name (or database unique name) used for search: DB1
channel ORA_DISK_1: AUTOBACKUP
/nfsfsxn/archlog/DB1/autobackup/2023_06_05/o1_mf_s_1138721482_l7vzyb
vq_.bkp found in the recovery area
channel ORA_DISK_1: looking for AUTOBACKUP on day: 20230607
channel ORA_DISK_1: looking for AUTOBACKUP on day: 20230606
channel ORA_DISK_1: looking for AUTOBACKUP on day: 20230605
channel ORA_DISK_1: restoring control file from AUTOBACKUP
/nfsfsxn/archlog/DB1/autobackup/2023_06_05/o1_mf_s_1138721482_l7vzyb
vq_.bkp
channel ORA_DISK_1: control file restore from AUTOBACKUP complete
output file name=/nfsfsxn/oracopy/db1.ctl
Finished restore at 07-JUN-23

RMAN> alter database mount;

released channel: ORA_DISK_1
Statement processed

RMAN> list incarnation;


List of Database Incarnations
DB Key  Inc Key DB Name  DB ID            STATUS  Reset SCN  Reset
Time
------- ------- -------- ---------------- --- ---------- ----------
1       1       DB1      1730530050       PARENT  1          17-APR-
19
2       2       DB1      1730530050       CURRENT 1920977    12-MAY-
23

RMAN> list copy of database tag 'OraCopyBKUPonFSxN_level_0';

List of Datafile Copies
=======================

Key     File S Completion Time Ckp SCN    Ckp Time        Sparse
------- ---- - --------------- ---------- --------------- ------
362     1    A 05-JUN-23       8319160    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSTEM_FNO-1_821tkrb8
        Tag: ORACOPYBKUPONFSXN_LEVEL_0

363     3    A 05-JUN-23       8319165    01-JUN-23       NO
```

```
        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_851tkrgf
        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_8h1tkrj9
        Tag: ORACOPYBKUPONFSXN_LEVEL_0

361     8    A 01-JUN-23       2383520    12-MAY-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-8_8g1tkrj7
        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_8a1tkrhr
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

376     10   A 05-JUN-23       8318714    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSAUX_FNO-10_861tkrgo
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

377     11   A 05-JUN-23       8318720    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
```

```
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_8e1tkril
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 4, PDB Name: DB1_PDB2

373     16   A 05-JUN-23       8319206    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
16_8j1tkrja
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 4, PDB Name: DB1_PDB2

369     17   A 05-JUN-23       8319188    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSTEM_FNO-17_8c1tkril
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 5, PDB Name: DB1_PDB3

367     18   A 05-JUN-23       8319180    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
SYSAUX_FNO-18_881tkrhr
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 5, PDB Name: DB1_PDB3
```

```
371     19   A 05-JUN-23       8319197    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-19_8f1tkrj4
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 5, PDB Name: DB1_PDB3

374     20   A 05-JUN-23       8319210    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
20_8k1tkrjb
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 5, PDB Name: DB1_PDB3

378     21   A 05-JUN-23       8318720    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
21_7j1tkqk6
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

388     22   A 05-JUN-23       8318714    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
22_7k1tkqk6
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

384     23   A 05-JUN-23       8318717    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
23_7l1tkqk6
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

389     24   A 05-JUN-23       8318719    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
24_7m1tkqk6
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

381     25   A 05-JUN-23       8318720    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
25_7n1tkqrh
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

392     26   A 05-JUN-23       8318714    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
26_7o1tkqrj
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
```

```
        Container ID: 3, PDB Name: DB1_PDB1

385     27   A 05-JUN-23       8318717    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
27_7p1tkqrq
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

390     28   A 05-JUN-23       8318719    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
28_7q1tkqs1
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

380     29   A 05-JUN-23       8318720    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
29_7r1tkr32
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

391     30   A 05-JUN-23       8318714    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
30_7s1tkr3a
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

382     31   A 05-JUN-23       8318717    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
31_7t1tkr3i
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

387     32   A 05-JUN-23       8318719    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
32_7u1tkr42
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

383     33   A 05-JUN-23       8318719    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
33_7v1tkra6
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

379     34   A 05-JUN-23       8318717    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
```

```
34_801tkram
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

386     35   A 05-JUN-23       8318714    01-JUN-23       NO
        Name: /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
35_811tkrap
        Tag: ORACOPYBKUPONFSXN_LEVEL_0
        Container ID: 3, PDB Name: DB1_PDB1

RMAN> switch database to copy;

datafile 1 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-1_821tkrb8"
datafile 3 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-3_831tkrd9"
datafile 4 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-4_851tkrgf"
datafile 5 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-5_8d1tkril"
datafile 6 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSAUX_FNO-6_891tkrhr"
datafile 7 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-USERS_FNO-7_8h1tkrj9"
datafile 8 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-UNDOTBS1_FNO-8_8g1tkrj7"
datafile 9 switched to datafile copy "/nfsfsxn/oracopy/data_D-DB1_I-
1730530050_TS-SYSTEM_FNO-9_8a1tkrhr"
datafile 10 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SYSAUX_FNO-10_861tkrgo"
datafile 11 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-UNDOTBS1_FNO-11_841tkrf2"
datafile 12 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-USERS_FNO-12_8i1tkrj9"
datafile 13 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SYSTEM_FNO-13_8b1tkril"
datafile 14 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SYSAUX_FNO-14_871tkrhr"
datafile 15 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-UNDOTBS1_FNO-15_8e1tkril"
datafile 16 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-USERS_FNO-16_8j1tkrja"
datafile 17 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SYSTEM_FNO-17_8c1tkril"
datafile 18 switched to datafile copy "/nfsfsxn/oracopy/data_D-
DB1_I-1730530050_TS-SYSAUX_FNO-18_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_l7vxylwp_.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_l7vyvrm5_.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_l7vyvrm5
_.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_7n1tkqrh
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-26_7o1tkqrj

NAME
--------------------------------------------------------------------
--------------------
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-27_7p1tkqrq
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-28_7q1tkqs1
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-29_7r1tkr32
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-30_7s1tkr3a
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-31_7t1tkr3i
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-32_7u1tkr42
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-33_7v1tkra6
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-34_801tkram
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-35_811tkrap
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-10_861tkrgo
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-14_871tkrhr

NAME
--------------------------------------------------------------------
--------------------
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-18_881tkrhr
```

```
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-3_831tkrd9
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-6_891tkrhr
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-13_8b1tkril
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-17_8c1tkril
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-1_821tkrb8
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-5_8d1tkril
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-9_8a1tkrhr
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-11_841tkrf2
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-15_8e1tkril
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-19_8f1tkrj4

NAME
--------------------------------------------------------------------
---------------------
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-4_851tkrgf
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-UNDOTBS1_FNO-8_8g1tkrj7
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-12_8i1tkrj9
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-16_8j1tkrja
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-20_8k1tkrjb
/nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-7_8h1tkrj9
/nfsfsxn/oracopy/db1.ctl
/nfsfsxn/oracopy/redo01.log
/nfsfsxn/oracopy/redo02.log
/nfsfsxn/oracopy/redo03.log

43 rows selected.

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> select * from test;

        ID DT
EVENT
----------
--------------------------------------------------------------------
-------
```

```
-----------------------------------------------------------------
         1 18-MAY-23 02.35.37.000000 PM
test oracle incremental merge switch to copy
         2 30-MAY-23 05.23.11.000000 PM
test recovery on a new EC2 instance host with image copy on FSxN
         3 05-JUN-23 03.19.46.000000 PM
test clone on a new EC2 instance host with image copy on FSxN

SQL>
```

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

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights
reserved.

Connected to database DB1 (DBID=1730530050)
```

```
Connected to server version 19.18.0

Control Files in database:
    /nfsfsxn/oracopy/db1.ctl

Change database ID and database name DB1 to DB1TST? (Y/[N]) => Y

Proceeding with operation
Changing database ID from 1730530050 to 3054879890
Changing database name from DB1 to DB1TST
    Control File /nfsfsxn/oracopy/db1.ctl - modified
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-
1_821tkrb - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-
3_831tkrd - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-4_851tkrg - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-
5_8d1tkri - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-
6_891tkrh - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
7_8h1tkrj - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-8_8g1tkrj - 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-
UNDOTBS1_FNO-11_841tkrf - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
12_8i1tkrj - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-
13_8b1tkri - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-
14_871tkrh - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
UNDOTBS1_FNO-15_8e1tkri - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
16_8j1tkrj - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSTEM_FNO-
17_8c1tkri - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SYSAUX_FNO-
18_881tkrh - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-
```

```
UNDOTBS1_FNO-19_8f1tkrj - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-USERS_FNO-
20_8k1tkrj - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
21_7j1tkqk - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
22_7k1tkqk - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
23_7l1tkqk - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
24_7m1tkqk - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
25_7n1tkqr - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
26_7o1tkqr - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
27_7p1tkqr - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
28_7q1tkqs - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
29_7r1tkr3 - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
30_7s1tkr3 - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
31_7t1tkr3 - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
32_7u1tkr4 - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
33_7v1tkra - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
34_801tkra - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/data_D-DB1_I-1730530050_TS-SOE_FNO-
35_811tkra - dbid changed, wrote new name
    Datafile /nfsfsxn/oracopy/DB1/datafile/o1_mf_temp_l81bhwjg_.tm -
dbid changed, wrote new name
    Datafile
/nfsfsxn/oracopy/DB1/FB864A929AEB79B9E053630F1EAC7046/datafile/o1_mf
_temp_l81bhz6g_.tm - dbid changed, wrote new name
    Datafile
/nfsfsxn/oracopy/DB1/FB867DA8C68C816EE053630F1EAC2BCF/datafile/o1_mf
_temp_l81bj16t_.tm - dbid changed, wrote new name
    Datafile
/nfsfsxn/oracopy/DB1/FB867EA89ECF81C0E053630F1EACB901/datafile/o1_mf
_temp_l81bj135_.tm - dbid changed, wrote new name
    Datafile
/nfsfsxn/oracopy/DB1/FB867F8A4D4F821CE053630F1EAC69CC/datafile/o1_mf
```

```
_temp_l81bj13g_.tm - dbid changed, wrote new name
    Control File /nfsfsxn/oracopy/db1.ctl - dbid changed, wrote new
name
    Instance shut down

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.
Succesfully changed database name and ID.
DBNEWID - Completed succesfully.
```

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
SQL>
```

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

  https://docs.oracle.com/en/database/oracle/oracle-database/19/bradv/getting-started-rman.html

- 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!467723097970!e!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARIsAJzSrdqwQrghn6I71jiWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:G:s&s_kwcid=AL!4422!3!467723097970!e!!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

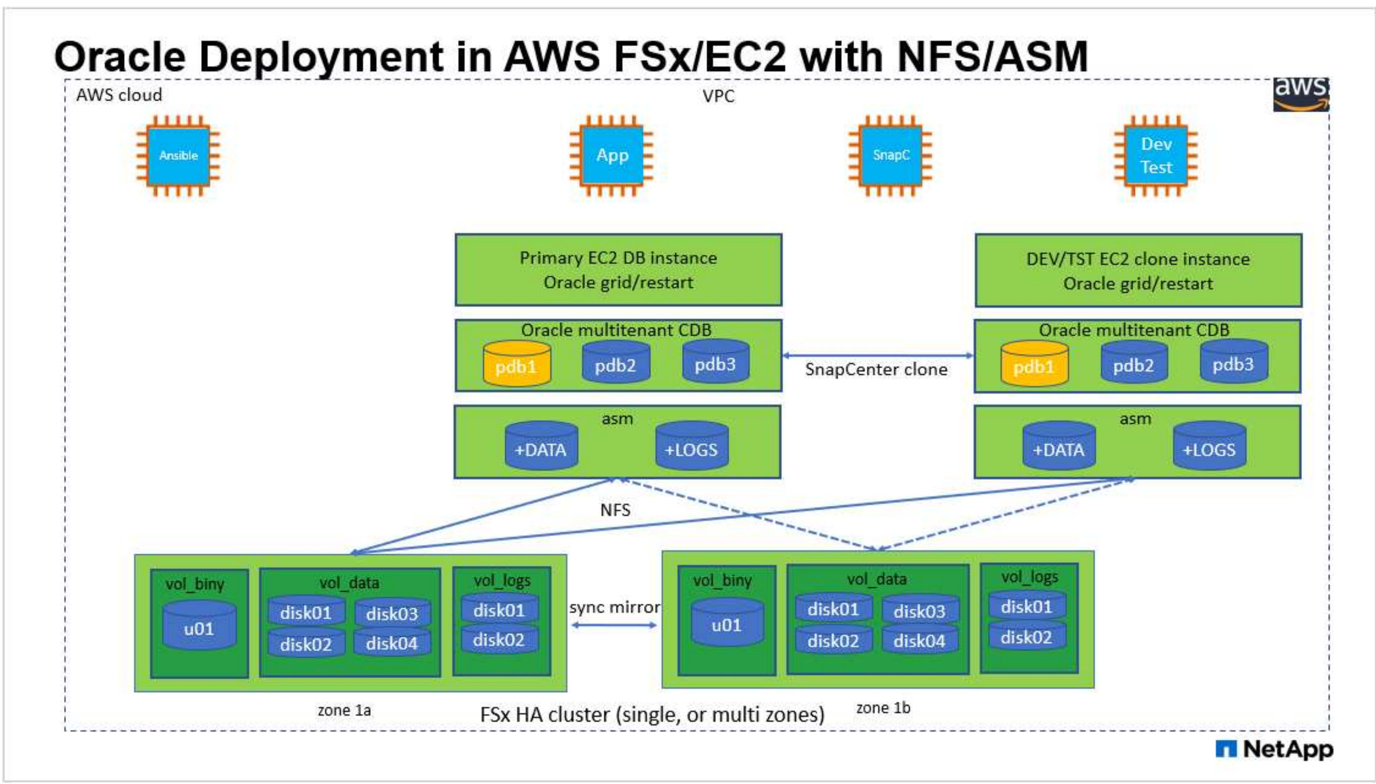
This solution is intended for the following people:

- 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




### Hardware and software components

**Hardware**

| | | |
|---|---|---|
| FSx ONTAP storage | Current version offered by AWS | One FSx HA cluster in the same VPC and availability zone |
| EC2 instance for compute | t2.xlarge/4vCPU/16G | Two EC2 T2 xlarge EC2 instances, one as primary DB server and the other as a clone DB server |
| | **Software** | |
| RedHat Linux | RHEL-8.6.0_HVM-20220503-x86_64-2-Hourly2-GP2 | Deployed RedHat subscription for testing |
| 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 Service | Version | v2.3.1.2324 |

**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 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.
- **Oracle data and logs layout.** In our tests and validations, we deployed two ASM disk groups for data and logs respectively. Within the +DATA asm disk group, we provisioned four disks in a data NFS file system mount point. Within the +LOGS asm disk group, we provisioned two disks in a logs NFS file system mount point. For large database deployment, ASM disk groups can be built to span multiple FSx file systems with ASM NFS disks distributed through multiple NFS mount points anchored on FSx file systems. This particular setup is designed to meet database throughput over 4GBps throughput and 160,000 raw SSD IOPS requirement.
- **dNFS configuration.** dNFS is built into Oracle kernel and is known to dramatically increase Oracle database performance when Oracle is deployed to NFS storage. dNFS is packaged into Oracle binary but is not turned on by default. It should be turned on for any Oracle database deployment on NFS. For multiple FSx file systems deployment for large database, dNFS multi-path should be properly configured.

- **Oracle ASM redundancy level to use for each Oracle ASM disk group that you create.** Because FSx already mirrors the storage on the FSx cluster level, you should `ONLY` use External Redundancy, which means that the option does not allow Oracle ASM to mirror the contents of the disk group. This is particularly important as NFS for Oracle database data storage requires HARD NFS mount option which is NOT desirable for mirroring ASM contents on the Oracle level.
- **Database backup.** NetApp provides a SaaS version of SnapCenter software service for database backup, restore, and clone in the cloud that is available through the NetApp BlueXP console UI. NetApp recommends implementing such a service to achieve fast (under a minute) SnapShot backup, quick (few minutes) database restore, and database cloning.

## Solution deployment

The following section provides step-by-step deployment procedures.

### Prerequisites for deployment

Deployment requires the following prerequisites.

1. An AWS account has been set up, and the necessary VPC and network segments have been created within your AWS account.
2. From the AWS EC2 console, you must deploy two EC2 Linux instances, one as the primary Oracle DB server and an optional alternative clone target DB server. See the architecture diagram in the previous section for more details about the environment setup. Also review the User Guide for Linux instances for more information.
3. From the AWS EC2 console, deploy Amazon FSx for ONTAP storage HA clusters to host the Oracle database volumes. If you are not familiar with the deployment of FSx storage, see the documentation Creating FSx for ONTAP file systems for step-by-step instructions.
4. Steps 2 and 3 can be performed using the following Terraform automation toolkit, which creates an EC2 instance named `ora_01` and an FSx file system named `fsx_01`. Review the instruction carefully and change the variables to suit your environment before execution.

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

Ensure that you have allocated at least 50G in EC2 instance root volume in order to have sufficient space to stage Oracle installation files.

### EC2 instance kernel configuration

With the prerequisites provisioned, log into the EC2 instance as ec2-user and sudo to root user to configure the Linux kernel for Oracle installation.

1. Create a staging directory `/tmp/archive` folder and set the `777` permission.

```
mkdir /tmp/archive

chmod 777 /tmp/archive
```

2. Download and stage the Oracle binary installation files and other required rpm files to the `/tmp/archive` directory.

   See the following list of installation files to be stated in `/tmp/archive` on the EC2 instance.

```
[ec2-user@ip-172-30-15-58 ~]$ ls -l /tmp/archive
total 10537316
-rw-rw-r--. 1 ec2-user ec2-user      19112 Mar 21 15:57 compat-
libcap1-1.10-7.el7.x86_64.rpm
-rw-rw-r--  1 ec2-user ec2-user 3059705302 Mar 21 22:01
LINUX.X64_193000_db_home.zip
-rw-rw-r--  1 ec2-user ec2-user 2889184573 Mar 21 21:09
LINUX.X64_193000_grid_home.zip
-rw-rw-r--. 1 ec2-user ec2-user     589145 Mar 21 15:56
netapp_linux_unified_host_utilities-7-1.x86_64.rpm
-rw-rw-r--. 1 ec2-user ec2-user      31828 Mar 21 15:55 oracle-
database-preinstall-19c-1.0-2.el8.x86_64.rpm
-rw-rw-r--  1 ec2-user ec2-user 2872741741 Mar 21 22:31
p34762026_190000_Linux-x86-64.zip
-rw-rw-r--  1 ec2-user ec2-user 1843577895 Mar 21 22:32
p34765931_190000_Linux-x86-64.zip
-rw-rw-r--  1 ec2-user ec2-user  124347218 Mar 21 22:33
p6880880_190000_Linux-x86-64.zip
-rw-r--r--  1 ec2-user ec2-user     257136 Mar 22 16:25
policycoreutils-python-utils-2.9-9.el8.noarch.rpm
```

3. Install Oracle 19c preinstall RPM, which satisfies most kernel configuration requirements.

```
yum install /tmp/archive/oracle-database-preinstall-19c-1.0-
2.el8.x86_64.rpm
```

4. Download and install the missing `compat-libcap1` in Linux 8.

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

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

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

6. Install `policycoreutils-python-utils`, which is not available in the EC2 instance.

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

7. Install open JDK version 1.8.

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

8. Install 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.

```
vi /etc/sysconfig/selinux
```

11. Add the following lines to `limit.conf` to set the file descriptor limit and stack size without quotes " ".

```
vi /etc/security/limits.conf
  "*               hard    nofile          65536"
  "*               soft    stack           10240"
```

12. Add swap space to EC2 instance by following this instruction: How do I allocate memory to work as swap space in an Amazon EC2 instance by using a swap file? The exact amount of space to add depends on the size of RAM up to 16G.
13. Add the ASM group to be used for the asm sysasm group

```
groupadd asm
```

14. Modify the oracle user to add ASM as a secondary group (the oracle user should have been created after Oracle preinstall RPM installation).

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

15. Reboot the EC2 instance.

**Provision and export NFS volumes to be mounted to EC2 instance host**

Provision three volumes from the command line by login to FSx cluster via ssh as fsxadmin user with FSx cluster management IP to host the Oracle database binary, data, and logs files.

1. Log into the FSx cluster through SSH as the fsxadmin user.

```
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:

```
FsxId02ad7bf3476b741df::> vol show
  (vol show)
FsxId06c3c8b2a7bd56458::> vol show
Vserver   Volume       Aggregate    State      Type       Size
Available Used%
--------- ------------ ------------ ---------- ---- ----------
---------- -----
svm_ora   ora_01_biny  aggr1        online     RW         50GB
47.50GB    0%
svm_ora   ora_01_data  aggr1        online     RW        100GB
95.00GB    0%
svm_ora   ora_01_logs  aggr1        online     RW        100GB
95.00GB    0%
svm_ora   svm_ora_root aggr1        online     RW          1GB
972.1MB    0%
4 entries were displayed.
```

**Database storage configuration**

Now, import and set up the FSx storage for the Oracle grid infrastructure and database installation on the EC2 instance host.

1. Log into the EC2 instance via SSH as the ec2-user with your SSH key and EC2 instance IP address.

```
ssh -i ora_01.pem ec2-user@172.30.15.58
```

2. Create /u01 directory to mount Oracle binary file system

```
sudo mkdir /u01
```

3. Mount the binary volume to `/u01`, changed to your FSx NFS lif IP address. If you deployed FSx cluster via NetApp automation toolkit, FSx virtual storage server NFS lif IP address will be listed in the output at the end of resources provision execution. Otherwise, it can be retrieved from AWS FSx console UI.

```
sudo mount -t nfs 172.30.15.19:/ora_01_biny /u01 -o
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536
```

4. Change `/u01` mount point ownership to the Oracle user and it's associated primary group.

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

5. Create /oradata directory to mount Oracle data file system

```
sudo mkdir /oradata
```

6. Mount the data volume to `/oradata`, changed to your FSx NFS lif IP address

```
sudo mount -t nfs 172.30.15.19:/ora_01_data /oradata -o
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536
```

7. Change `/oradata` mount point ownership to the Oracle user and it's associated primary group.

```
sudo chown oracle:oinstall /oradata
```

8. Create /oralogs directory to mount Oracle logs file system

```
sudo mkdir /oralogs
```

9. Mount the log volume to `/oralogs`, changed to your FSx NFS lif IP address

```
sudo mount -t nfs 172.30.15.19:/ora_01_logs /oralogs -o
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536
```

10. Change `/oralogs` mount point ownership to the Oracle user and it's associated primary group.

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

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

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

Add the following line.

```
172.30.15.19:/ora_01_biny       /u01            nfs
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536   0
0
172.30.15.19:/ora_01_data       /oradata        nfs
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536   0
0
172.30.15.19:/ora_01_logs       /oralogs        nfs
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536   0
0
```

12. sudo to oracle user, create asm folders to store asm disk files

```
sudo su
su - oracle
mkdir /oradata/asm
mkdir /oralogs/asm
```

13. As the oracle user, create asm data disk files, change the count to match to the disk size with block size.

```
dd if=/dev/zero of=/oradata/asm/nfs_data_disk01 bs=1M count=20480
oflag=direct
dd if=/dev/zero of=/oradata/asm/nfs_data_disk02 bs=1M count=20480
oflag=direct
dd if=/dev/zero of=/oradata/asm/nfs_data_disk03 bs=1M count=20480
oflag=direct
dd if=/dev/zero of=/oradata/asm/nfs_data_disk04 bs=1M count=20480
oflag=direct
```

14. As the root user, change data disk file permission to 640

```
chmod 640 /oradata/asm/*
```

15. AS the oracle user, create asm logs disk files, change to count to match to the disk size with block size.

```
dd if=/dev/zero of=/oralogs/asm/nfs_logs_disk01 bs=1M count=40960
oflag=direct
dd if=/dev/zero of=/oralogs/asm/nfs_logs_disk02 bs=1M count=40960
oflag=direct
```

16. As the root user, change logs disk file permission to 640

```
chmod 640 /oralogs/asm/*
```

17. Reboot the EC2 instance host.

#### Oracle grid infrastructure installation

1. Log into the EC2 instance as the ec2-user via SSH and enable password authentication by uncommenting `PasswordAuthentication yes` and then commenting out `PasswordAuthentication no`.

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

2. Restart the sshd service.

   ```
   sudo systemctl restart sshd
   ```

3. Reset the Oracle user password.

   ```
   sudo passwd oracle
   ```

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

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

5. Change the directory permission setting.

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

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

   ```
   mkdir -p /u01/app/oracle/product/19.0.0/grid
   cd /u01/app/oracle/product/19.0.0/grid
   ```

7. Unzip the grid installation files.

   ```
   unzip -q /tmp/archive/LINUX.X64_193000_grid_home.zip
   ```

8. From grid home, delete the `OPatch` directory.

   ```
   rm -rf OPatch
   ```

9. From grid home, copy `p6880880_190000_Linux-x86-64.zip` to the grid_home, and then unzip it.

```
cp /tmp/archive/p6880880_190000_Linux-x86-64.zip .
unzip p6880880_190000_Linux-x86-64.zip
```

10. From grid home, revise `cv/admin/cvu_config`, uncomment and replace `CV_ASSUME_DISTID=OEL5` with `CV_ASSUME_DISTID=OL7`.

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

11. Prepare a `gridsetup.rsp` file for silent installation and place the rsp file in the `/tmp/archive` directory. The rsp file should cover sections A, B, and G with the following infomation:

```
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_da
ta_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.

```
 ./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.

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
+
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
```

#### 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, copy `p6880880_190000_Linux-x86-64.zip` to `grid_home`, and then unzip it.

```
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`.

```
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:

```
oracle.install.option=INSTALL_DB_SWONLY
UNIX_GROUP_NAME=oinstall
INVENTORY_LOCATION=/u01/app/oraInventory
ORACLE_HOME=/u01/app/oracle/product/19.0.0/db1
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 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 sofware-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, lauch 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
      1        ONLINE  ONLINE       ip-172-30-15-58          STABLE
ora.db1.db
      1        ONLINE  ONLINE       ip-172-30-15-58
Open,HOME=/u01/app/o

racle/product/19.0.0

/db1,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]$
```

14. Set the Oracle user `.bash_profile`.

```
vi ~/.bash_profile
```

15. Add following entries:

```
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db1
export ORACLE_SID=db1
export PATH=$PATH:$ORACLE_HOME/bin
alias asm='export
ORACLE_HOME=/u01/app/oracle/product/19.0.0/grid;export
ORACLE_SID=+ASM;export PATH=$PATH:$ORACLE_HOME/bin'
```

16. Validate the CDB/PDB created.

```
. ~/.bash_profile

sqlplus / as sysdba

SQL> select name, open_mode from v$database;

NAME      OPEN_MODE

DB1       READ WRITE

SQL> select name from v$datafile;

NAME

+DATA/DB1/DATAFILE/system.256.1132176177
+DATA/DB1/DATAFILE/sysaux.257.1132176221
+DATA/DB1/DATAFILE/undotbs1.258.1132176247
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/system.265.11321
77009
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/sysaux.266.11321
77009
+DATA/DB1/DATAFILE/users.259.1132176247
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/undotbs1.267.113
2177009
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/system.271.11321
77853
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/sysaux.272.11321
77853
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/undotbs1.270.113
2177853
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/users.274.113217
```

```
7871

NAME

+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/system.276.11321
77871
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/sysaux.277.11321
77871
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/undotbs1.275.113
2177871
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/users.279.113217
7889
+DATA/DB1/F78529A14DD8BB18E0533A0F1EACB8ED/DATAFILE/system.281.11321
77889
+DATA/DB1/F78529A14DD8BB18E0533A0F1EACB8ED/DATAFILE/sysaux.282.11321
77889
+DATA/DB1/F78529A14DD8BB18E0533A0F1EACB8ED/DATAFILE/undotbs1.280.113
2177889
+DATA/DB1/F78529A14DD8BB18E0533A0F1EACB8ED/DATAFILE/users.284.113217
7907

19 rows selected.

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>
```

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
--------------------------------------------------------------------
------------
DIRNAME
--------------------------------------------------------------------
------------
fsx_01
/ora_01_data

fsx_01
/ora_01_biny

fsx_01
/ora_01_logs
```

22. Validate Oracle ASM

```
[oracle@ip-172-30-15-58 db1]$ asm
[oracle@ip-172-30-15-58 db1]$ sqlplus / as sysasm

SQL*Plus: Release 19.0.0.0.0 - Production on Tue May 9 20:39:39 2023
Version 19.18.0.0.0

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


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0

SQL> set lin 200
SQL> col path form a30
SQL> select name, path, header_status, mount_status, state from
v$asm_disk;

NAME                           PATH
```

```
HEADER_STATU MOUNT_S STATE
------------------------------ ------------------------------
------------ ------- --------
DATA_0002                      /oradata/asm/nfs_data_disk01   MEMBER
CACHED  NORMAL
DATA_0000                      /oradata/asm/nfs_data_disk02   MEMBER
CACHED  NORMAL
DATA_0001                      /oradata/asm/nfs_data_disk03   MEMBER
CACHED  NORMAL
DATA_0003                      /oradata/asm/nfs_data_disk04   MEMBER
CACHED  NORMAL
LOGS_0000                      /oralogs/asm/nfs_logs_disk01   MEMBER
CACHED  NORMAL
LOGS_0001                      /oralogs/asm/nfs_logs_disk02   MEMBER
CACHED  NORMAL

6 rows selected.


SQL> select name, state, ALLOCATION_UNIT_SIZE, TOTAL_MB, FREE_MB
from v$asm_diskgroup;

NAME                           STATE       ALLOCATION_UNIT_SIZE
TOTAL_MB    FREE_MB
------------------------------ ----------- --------------------
---------- ----------
DATA                           MOUNTED                  4194304
81920      73536
LOGS                           MOUNTED                  4194304
81920      81640

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

  https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/installing-oracle-grid-infrastructure-for-a-standalone-server-with-a-new-database-installation.html#GUID-0B1CEE8C-C893-46AA-8A6A-7B5FAAEC72B3

- Installing and Configuring Oracle Database Using Response Files

  https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/installing-and-configuring-oracle-database-using-response-files.html#GUID-D53355E9-E901-4224-9A2A-B882070EDDF7

- 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!467723097970!e!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARIsAJzSrdqwQrghn6I71jiWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:G:s&s_kwcid=AL!4422!3!467723097970!e!!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 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.

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

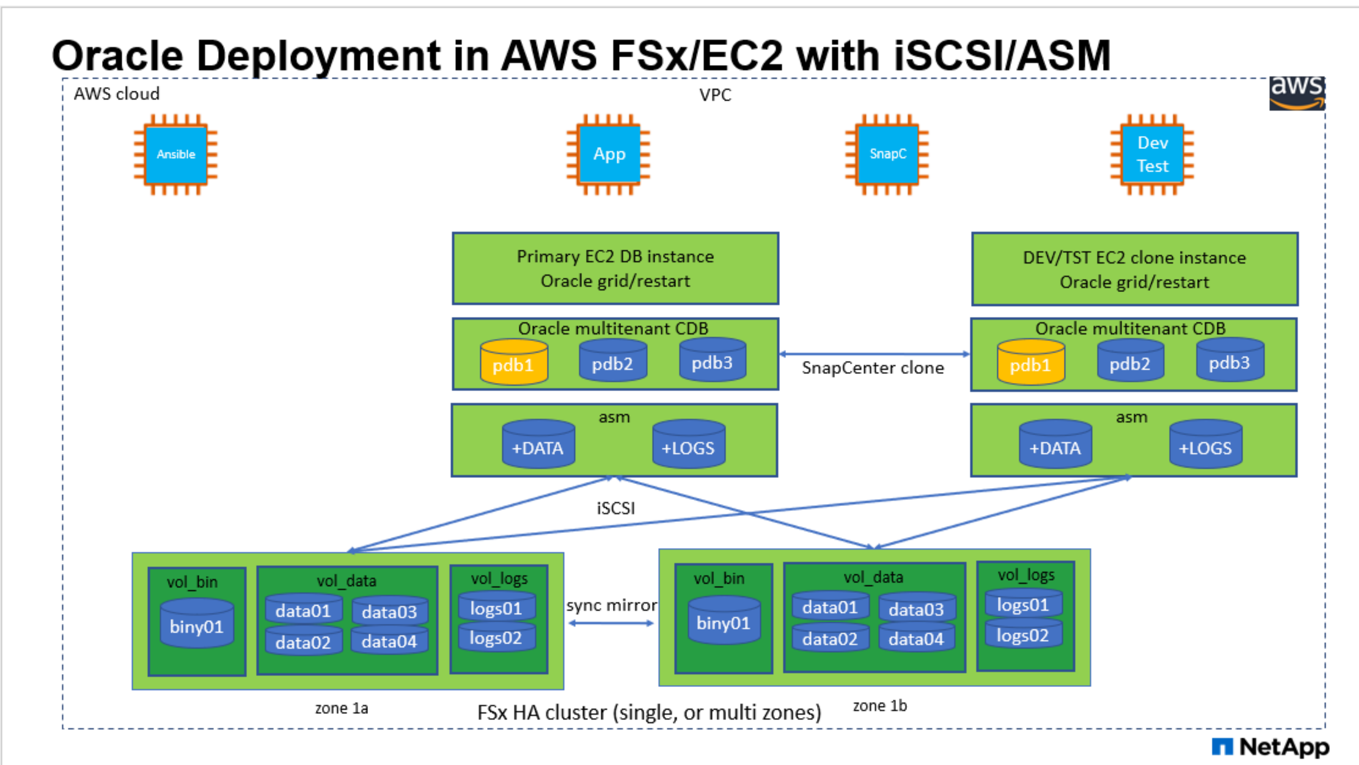
This solution is intended for the following people:

* 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




### Hardware and software components

| Hardware | | |
|---|---|---|
| FSx ONTAP storage | Current version offered by AWS | One FSx HA cluster in the same VPC and availability zone |

| | | |
|---|---|---|
| EC2 instance for compute | t2.xlarge/4vCPU/16G | Two EC2 T2 xlarge EC2 instances, one as primary DB server and the other as a clone DB server |
| | **Software** | |
| RedHat Linux | RHEL-8.6.0_HVM-20220503-x86_64-2-Hourly2-GP2 | Deployed RedHat subscription for testing |
| 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 Service | Version | v2.3.1.2324 |

**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 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.
- **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 peek 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.

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

Ensure that you have allocated at least 50G in EC2 instance root volume in order to have sufficient space to stage Oracle installation files.

### EC2 instance kernel configuration

With the prerequisites provisioned, log into the EC2 instance as ec2-user and sudo to root user to configure the Linux kernel for Oracle installation.

1. Create a staging directory `/tmp/archive` folder and set the `777` permission.

```
mkdir /tmp/archive

chmod 777 /tmp/archive
```

2. Download and stage the Oracle binary installation files and other required rpm files to the `/tmp/archive` directory.

   See the following list of installation files to be stated in `/tmp/archive` on the EC2 instance.

```
[ec2-user@ip-172-30-15-58 ~]$ ls -l /tmp/archive
total 10537316
-rw-rw-r--. 1 ec2-user ec2-user      19112 Mar 21 15:57 compat-
libcap1-1.10-7.el7.x86_64.rpm
-rw-rw-r--  1 ec2-user ec2-user 3059705302 Mar 21 22:01
LINUX.X64_193000_db_home.zip
-rw-rw-r--  1 ec2-user ec2-user 2889184573 Mar 21 21:09
LINUX.X64_193000_grid_home.zip
-rw-rw-r--. 1 ec2-user ec2-user     589145 Mar 21 15:56
netapp_linux_unified_host_utilities-7-1.x86_64.rpm
-rw-rw-r--. 1 ec2-user ec2-user      31828 Mar 21 15:55 oracle-
database-preinstall-19c-1.0-2.el8.x86_64.rpm
-rw-rw-r--  1 ec2-user ec2-user 2872741741 Mar 21 22:31
p34762026_190000_Linux-x86-64.zip
-rw-rw-r--  1 ec2-user ec2-user 1843577895 Mar 21 22:32
p34765931_190000_Linux-x86-64.zip
-rw-rw-r--  1 ec2-user ec2-user  124347218 Mar 21 22:33
p6880880_190000_Linux-x86-64.zip
-rw-r--r--  1 ec2-user ec2-user     257136 Mar 22 16:25
policycoreutils-python-utils-2.9-9.el8.noarch.rpm
```

3. Install Oracle 19c preinstall RPM, which satisfies most kernel configuration requirements.

```
yum install /tmp/archive/oracle-database-preinstall-19c-1.0-
2.el8.x86_64.rpm
```

4. Download and install the missing `compat-libcap1` in Linux 8.

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

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

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

6. Install `policycoreutils-python-utils`, which is not available in the EC2 instance.

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

7. Install open JDK version 1.8.

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

8. Install iSCSI initiator utils.

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

9. Install `sg3_utils`.

```
yum install sg3_utils
```

10. Install `device-mapper-multipath`.

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

11. Disable transparent hugepages in the current system.

```
echo never > /sys/kernel/mm/transparent_hugepage/enabled
echo never > /sys/kernel/mm/transparent_hugepage/defrag
```

Add the following lines in `/etc/rc.local` to disable `transparent_hugepage` after reboot:

```
  # 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.timeo.replacement_timeout` in the `iscsi.conf` configuration file from 120 to 5 seconds.

```
vi /etc/iscsi/iscsid.conf
```

16. Enable and start the iSCSI service on the EC2 instance.

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

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

```
cat /etc/iscsi/initiatorname.iscsi
```

18. Add the ASM group to be used for the asm sysasm group

```
groupadd asm
```

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. Reboot the EC2 instance.

**Provision and map database volumes and LUNs to the EC2 instance host**

Provision three volumes from the command line by login to FSx cluster via ssh as fsxadmin user with FSx cluster management IP to host the Oracle database binary, data, and logs files.

1. Log into the FSx cluster through SSH as the fsxadmin user.

   ```
   ssh fsxadmin@172.30.15.53
   ```

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

   ```
   vol create -volume ora_01_biny -aggregate aggr1 -size 50G -state
   online  -type RW -snapshot-policy none -tiering-policy snapshot-only
   ```

3. Execute the following command to create a volume for Oracle data.

   ```
   vol create -volume ora_01_data -aggregate aggr1 -size 100G -state
   online  -type RW -snapshot-policy none -tiering-policy snapshot-only
   ```

4. Execute the following command to create a volume for Oracle logs.

   ```
   vol create -volume ora_01_logs -aggregate aggr1 -size 100G -state
   online  -type RW -snapshot-policy none -tiering-policy snapshot-only
   ```

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

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

6. Create data LUNs within the database data volume.

   ```
   lun create -path /vol/ora_01_data/ora_01_data_01 -size 20G -ostype
   linux

   lun create -path /vol/ora_01_data/ora_01_data_02 -size 20G -ostype
   linux

   lun create -path /vol/ora_01_data/ora_01_data_03 -size 20G -ostype
   linux

   lun create -path /vol/ora_01_data/ora_01_data_04 -size 20G -ostype
   linux
   ```

7. Create log LUNs within the database logs volume.

```
lun create -path /vol/ora_01_logs/ora_01_logs_01 -size 40G -ostype
linux

lun create -path /vol/ora_01_logs/ora_01_logs_02 -size 40G -ostype
linux
```

8. Create an igroup for the EC2 instance with the initiator retrieved from step 14 of the EC2 kernel configuration above.

```
igroup create -igroup ora_01 -protocol iscsi -ostype linux
-initiator iqn.1994-05.com.redhat:f65fed7641c2
```

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:

```
FsxId02ad7bf3476b741df::> mapping show
  (lun mapping show)
Vserver    Path                                      Igroup   LUN ID
Protocol
---------- ----------------------------------------  -------  ------
--------
svm_ora    /vol/ora_01_biny/ora_01_biny_01           ora_01        0
iscsi
svm_ora    /vol/ora_01_data/ora_01_data_01           ora_01        1
iscsi
svm_ora    /vol/ora_01_data/ora_01_data_02           ora_01        2
iscsi
svm_ora    /vol/ora_01_data/ora_01_data_03           ora_01        3
iscsi
svm_ora    /vol/ora_01_data/ora_01_data_04           ora_01        4
iscsi
svm_ora    /vol/ora_01_logs/ora_01_logs_01           ora_01        5
iscsi
svm_ora    /vol/ora_01_logs/ora_01_logs_02           ora_01        6
iscsi
```

#### Database storage configuration

Now, import and set up the FSx storage for the Oracle grid infrastructure and database installation on the EC2 instance host.

1. Log into the EC2 instance via SSH as the ec2-user with your SSH key and EC2 instance IP address.

```
ssh -i ora_01.pem ec2-user@172.30.15.58
```

2. Discover the FSx iSCSI endpoints using either SVM iSCSI IP address. Then change to your environment-specific portal address.

```
sudo iscsiadm 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
Logging in to [iface: default, target: iqn.1992-
08.com.netapp:sn.1f795e65c74911edb785affbf0a2b26e:vs.3, portal:
172.30.15.51,3260]
Logging in to [iface: default, target: iqn.1992-
08.com.netapp:sn.1f795e65c74911edb785affbf0a2b26e:vs.3, portal:
172.30.15.13,3260]
Login to [iface: default, target: iqn.1992-
08.com.netapp:sn.1f795e65c74911edb785affbf0a2b26e:vs.3, portal:
172.30.15.51,3260] successful.
Login to [iface: default, target: iqn.1992-
08.com.netapp:sn.1f795e65c74911edb785affbf0a2b26e:vs.3, portal:
172.30.15.13,3260] successful.
```

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] 172.30.15.51:3260,1028 iqn.1992-
08.com.netapp:sn.1f795e65c74911edb785affbf0a2b26e:vs.3 (non-flash)
tcp: [2] 172.30.15.13:3260,1029 iqn.1992-
08.com.netapp:sn.1f795e65c74911edb785affbf0a2b26e: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.

```
[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/sdi        host3      iSCSI      20g     cDOT
svm_ora                       /vol/ora_01_data/ora_01_data_02
/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_data/ora_01_data_01
/dev/sdb        host2      iSCSI      20g     cDOT
svm_ora                       /vol/ora_01_biny/ora_01_biny_01
/dev/sda        host2      iSCSI      40g     cDOT
```

6. Configure the `multipath.conf` file with following default and blacklist entries.

```
sudo vi /etc/multipath.conf

defaults {
    find_multipaths yes
    user_friendly_names yes
}

blacklist {
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
    devnode "^hd[a-z]"
    devnode "^cciss.*"
}
```

7. Start the multipath service.

```
sudo systemctl start multipathd
```

Now multipath devices appear in the `/dev/mapper` directory.

```
[ec2-user@ip-172-30-15-58 ~]$ ls -l /dev/mapper
total 0
lrwxrwxrwx 1 root root       7 Mar 21 20:13
3600a09806c574235472455534e68512d -> ../dm-0
lrwxrwxrwx 1 root root       7 Mar 21 20:13
3600a09806c574235472455534e685141 -> ../dm-1
lrwxrwxrwx 1 root root       7 Mar 21 20:13
3600a09806c574235472455534e685142 -> ../dm-2
lrwxrwxrwx 1 root root       7 Mar 21 20:13
3600a09806c574235472455534e685143 -> ../dm-3
lrwxrwxrwx 1 root root       7 Mar 21 20:13
3600a09806c574235472455534e685144 -> ../dm-4
lrwxrwxrwx 1 root root       7 Mar 21 20:13
3600a09806c574235472455534e685145 -> ../dm-5
lrwxrwxrwx 1 root root       7 Mar 21 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:

```
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.

```
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:

```
[ec2-user@ip-172-30-15-58 ~]$ ls -l /dev/mapper
total 0
crw------- 1 root root 10, 236 Mar 21 18:19 control
lrwxrwxrwx 1 root root       7 Mar 21 20:41 ora_01_biny_01 -> ../dm-
0
lrwxrwxrwx 1 root root       7 Mar 21 20:41 ora_01_data_01 -> ../dm-
1
lrwxrwxrwx 1 root root       7 Mar 21 20:41 ora_01_data_02 -> ../dm-
2
lrwxrwxrwx 1 root root       7 Mar 21 20:41 ora_01_data_03 -> ../dm-
3
lrwxrwxrwx 1 root root       7 Mar 21 20:41 ora_01_data_04 -> ../dm-
4
lrwxrwxrwx 1 root root       7 Mar 21 20:41 ora_01_logs_01 -> ../dm-
5
lrwxrwxrwx 1 root root       7 Mar 21 20:41 ora_01_logs_02 -> ../dm-
6
```

11. Partition the binary LUN with a single primary partition.

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

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

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

13. Mount the binary LUN to `/u01`.

```
sudo mount -t xfs /dev/mapper/ora_01_biny_01p1 /u01
```

14. Change `/u01` mount point ownership to the Oracle user and it’s asssociated primary group.

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

15. Find the UUI of the binary LUN.

```
sudo blkid /dev/mapper/ora_01_biny_01p1
```

16. Add a mount point to `/etc/fstab`.

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

Add the following line.

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

> It is important to mount the binary with only the UUID and with the nofail option to avoid possible root-lock issues during EC2-instance reboot.

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

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

Include following entries:

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

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

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

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

```
udevadm trigger
```

20. As the root user, reload multipathd.

```
systemctl restart multipathd
```

21. Reboot the EC2 instance host.

#### Oracle grid infrastructure installation

1. Log into the EC2 instance as the ec2-user via SSH and enable password authentication by uncommenting `PasswordAuthentication yes` and then commenting out `PasswordAuthentication no`.

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

2. Restart the sshd service.

   ```
   sudo systemctl restart sshd
   ```

3. Reset the Oracle user password.

   ```
   sudo passwd oracle
   ```

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

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

5. Change the directory permission setting.

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

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

   ```
   mkdir -p /u01/app/oracle/product/19.0.0/grid
   cd /u01/app/oracle/product/19.0.0/grid
   ```

7. Unzip the grid installation files.

   ```
   unzip -q /tmp/archive/LINUX.X64_193000_grid_home.zip
   ```

8. From grid home, delete the `OPatch` directory.

   ```
   rm -rf OPatch
   ```

9. From grid home, unzip `p6880880_190000_Linux-x86-64.zip`.

```
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`.

```
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 infomation:

```
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/m
apper/ora_01_data_02,/dev/mapper/ora_01_data_03,/dev/mapper/ora_01_d
ata_04
oracle.install.asm.diskGroup.diskDiscoveryString=/dev/mapper/*
oracle.install.asm.monitorPassword="SetPWD"
oracle.install.asm.configureAFD=true
```

12. Log into the EC2 instance as the root user and set `ORACLE_HOME` and `ORACLE_BASE`.

```
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/grid
export ORACLE_BASE=/tmp
cd /u01/app/oracle/product/19.0.0/grid/bin
```

13. Provision disk devices for use with the Oracle ASM filter driver.

```
./asmcmd afd_label DATA01 /dev/mapper/ora_01_data_01 --init

./asmcmd afd_label DATA02 /dev/mapper/ora_01_data_02 --init

./asmcmd afd_label DATA03 /dev/mapper/ora_01_data_03 --init

./asmcmd afd_label DATA04 /dev/mapper/ora_01_data_04 --init

./asmcmd afd_label LOGS01 /dev/mapper/ora_01_logs_01 --init

./asmcmd afd_label LOGS02 /dev/mapper/ora_01_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 p34762026_190000_Linux-x86-64.zip
```

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:

```
/u01/app/oracle/product/19.0.0/grid/gridSetup.sh -executeConfigTools
-responseFile /tmp/archive/gridsetup.rsp -silent
```

21. As the Oracle user, create the LOGS disk group.

```
bin/asmca -silent -sysAsmPassword 'yourPWD' -asmsnmpPassword
'yourPWD' -createDiskGroup -diskGroupName LOGS -disk 'AFD:LOGS*'
-redundancy EXTERNAL -au_size 4
```

22. As the Oracle user, validate grid services after installation configuration.

```
bin/crsctl stat res -t
+
Name                Target  State        Server
State details
Local Resources
ora.DATA.dg         ONLINE  ONLINE       ip-172-30-15-58
STABLE
ora.LISTENER.lsnr   ONLINE  ONLINE       ip-172-30-15-58
STABLE
ora.LOGS.dg         ONLINE  ONLINE       ip-172-30-15-58
STABLE
ora.asm             ONLINE  ONLINE       ip-172-30-15-58
Started,STABLE
ora.ons             OFFLINE OFFLINE      ip-172-30-15-58
STABLE
Cluster Resources
ora.cssd            ONLINE  ONLINE       ip-172-30-15-58
STABLE
ora.diskmon         OFFLINE OFFLINE
STABLE
ora.driver.afd      ONLINE  ONLINE       ip-172-30-15-58
STABLE
ora.evmd            ONLINE  ONLINE       ip-172-30-15-58
STABLE
```

23. Valiate 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:

```
oracle.install.option=INSTALL_DB_SWONLY
UNIX_GROUP_NAME=oinstall
INVENTORY_LOCATION=/u01/app/oraInventory
ORACLE_HOME=/u01/app/oracle/product/19.0.0/db1
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 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 sofware-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
```

12. As Oracle user, lauch 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`.

```
vi ~/.bash_profile
```

15. Add following entries:

```
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db1
export ORACLE_SID=db1
export PATH=$PATH:$ORACLE_HOME/bin
alias asm='export
ORACLE_HOME=/u01/app/oracle/product/19.0.0/grid;export
ORACLE_SID=+ASM;export PATH=$PATH:$ORACLE_HOME/bin'
```

16. Validate the CDB/PDB created.

```
/home/oracle/.bash_profile

sqlplus / as sysdba
```

```
SQL> select name, open_mode from v$database;

NAME      OPEN_MODE

DB1       READ WRITE

SQL> select name from v$datafile;

NAME

+DATA/DB1/DATAFILE/system.256.1132176177
+DATA/DB1/DATAFILE/sysaux.257.1132176221
+DATA/DB1/DATAFILE/undotbs1.258.1132176247
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/system.265.11321
77009
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/sysaux.266.11321
77009
+DATA/DB1/DATAFILE/users.259.1132176247
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/undotbs1.267.113
2177009
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/system.271.11321
77853
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/sysaux.272.11321
77853
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/undotbs1.270.113
2177853
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/users.274.113217
7871

NAME

+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/system.276.11321
77871
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/sysaux.277.11321
77871
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/undotbs1.275.113
2177871
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/users.279.113217
7889
+DATA/DB1/F78529A14DD8BB18E0533A0F1EACB8ED/DATAFILE/system.281.11321
77889
+DATA/DB1/F78529A14DD8BB18E0533A0F1EACB8ED/DATAFILE/sysaux.282.11321
77889
+DATA/DB1/F78529A14DD8BB18E0533A0F1EACB8ED/DATAFILE/undotbs1.280.113
2177889
```

```
+DATA/DB1/F78529A14DD8BB18E0533A0F1EACB8ED/DATAFILE/users.284.113217
7907

19 rows selected.

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>
```

17. Set the DB recovery location to the +LOGS disk group.

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

alter system set db_recovery_file_dest = '+LOGS' scope=both;
```

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

```
sqlplus /as sysdba.

shutdown immediate;

startup mount;

alter database archivelog;

alter database open;
```

This completes Oracle 19c version 19.18 Restart deployment on an Amazon FSx for ONTAP and EC2 compute instance. If desired, NetApp recommends relocating the Oracle control file and online log files to the +LOGS disk group.

### Automated deployment option

NetApp will release a fully automated solution deployment toolkit with Ansible to facilitate the implementation of this solution. Please check back for the availability of the toolkit. After it is released, a link will be posted here.

### Oracle Database backup, restore, and clone with SnapCenter Service

See SnapCenter Services for Oracle for details on Oracle database backup, restore, and clone with NetApp BlueXP console.

### Where to find additional information

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

- Installing Oracle Grid Infrastructure for a Standalone Server with a New Database Installation

  https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/installing-oracle-grid-infrastructure-for-a-standalone-server-with-a-new-database-installation.html#GUID-0B1CEE8C-C893-46AA-8A6A-7B5FAAEC72B3

- Installing and Configuring Oracle Database Using Response Files

  https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/installing-and-configuring-oracle-database-using-response-files.html#GUID-D53355E9-E901-4224-9A2A-B882070EDDF7

- 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!467723097970!e!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARIsAJzSrdqwQrghn6I71jiWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:G:s&s_kwcid=AL!4422!3!467723097970!e!!g!!aws%20ec2

# Oracle Database Deployment on AWS EC2 and FSx Best Practices

## WP-7357: Oracle Database Deployment on EC2 and FSx Best Practices Introduction

Allen Cao, Niyaz Mohamed, Jeffrey Steiner, NetApp

Many mission-critical enterprise Oracle databases are still hosted on-premises, and many enterprises are looking to migrate these Oracle databases to a public cloud. Often, these Oracle databases are application centric and thus require user-specific configurations, a capability that is missing from many database-as-a-service public-cloud offerings. Therefore, the current database landscape calls for a public-cloud-based Oracle database solution built from a high-performance, scalable compute and storage service that can accommodate unique requirements. AWS EC2 compute instances and the AWS FSx storage service might be the missing pieces of this puzzle that you can leverage to build and migrate your mission critical Oracle database workloads to a public cloud.

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for enterprises. The simple Amazon EC2 web-service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment.

Amazon FSx for ONTAP is an AWS storage service that uses industry-leading NetApp ONTAP block and file storage, which exposes NFS, SMB, and iSCSI. With such a powerful storage engine, it has never been easier to relocate mission-critical Oracle database apps to AWS with sub-millisecond response times, multiple GBps of throughput, and 100,000+ IOPS per database instance. Better yet, the FSx storage service comes with native replication capability that allows you to easily migrate your on-premises Oracle database to AWS or to replicate your mission critical Oracle database to a secondary AWS availability zone for HA or DR.

The goal of this documentation is to provide step-by-step processes, procedures, and best-practice guidance on how to deploy and configure an Oracle database with FSx storage and an EC2 instance that delivers performance similar to an on-premises system. NetApp also provides an automation toolkit that automates most of the tasks that are required for the deployment, configuration, and management of your Oracle database workload in the AWS public cloud.

To learn more about the solution and use case, take a look at following overview video:

Modernize your Oracle database with hybrid cloud in AWS and FSx ONTAP, Part1 - Use case and solution architecture

Next: Solutions architecture.

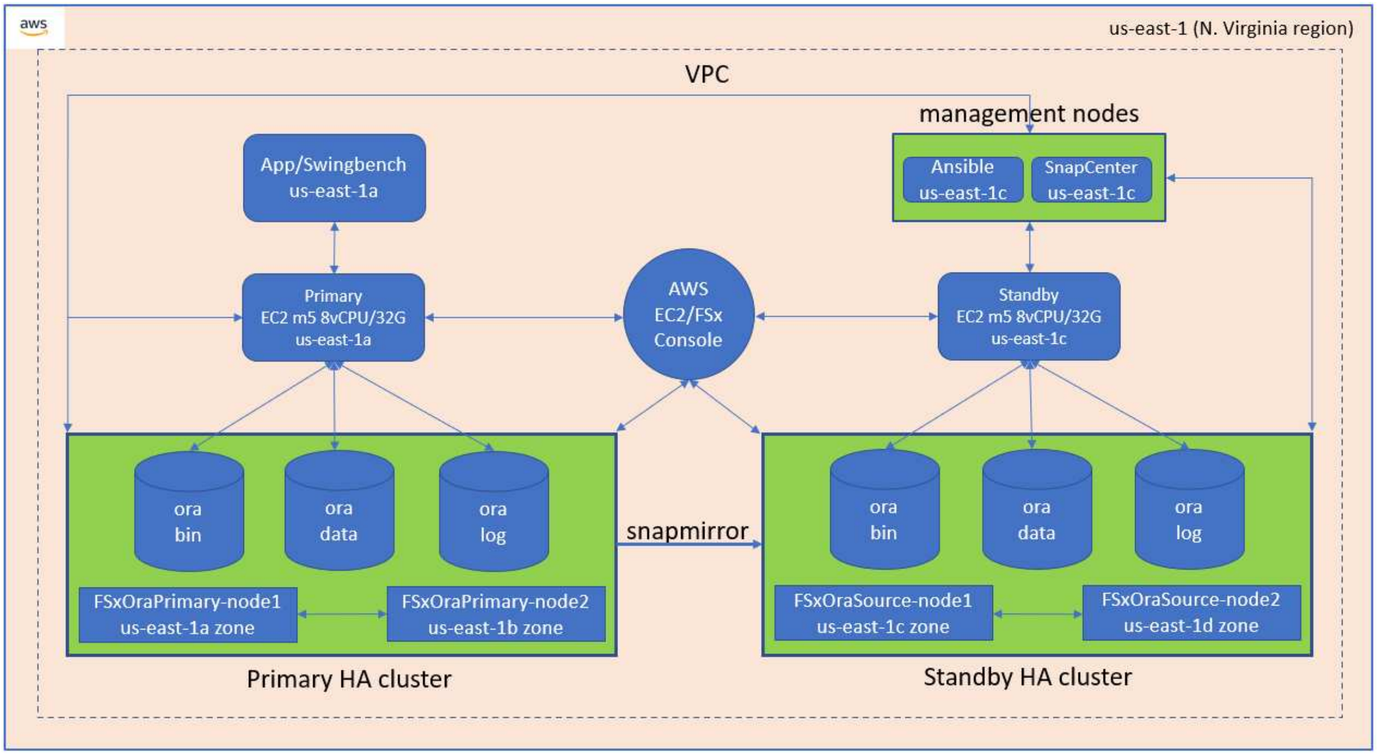
## Solution architecture

Previous: Introduction.

### 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.

Next: Factors to consider.

## Factors to consider for Oracle database deployment

Previous: Solution architecture.

A public cloud provides many choices for compute and storage, and using the correct type of compute instance and storage engine is a good place to start for database deployment. You should also select compute and storage configurations that are optimized for Oracle databases.

The following sections describe the key considerations when deploying Oracle database in an AWS public cloud on an EC2 instance with FSx storage.

### VM performance

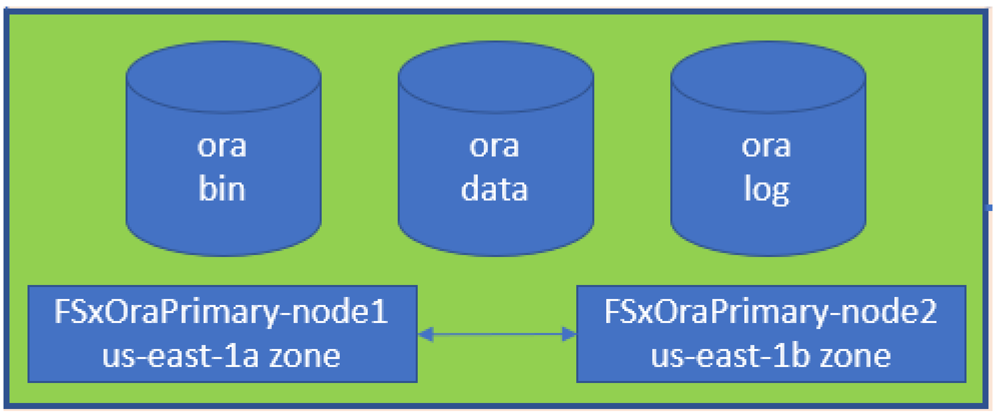
Selecting the right VM size is important for optimal performance of a relational database in a public cloud. For better performance, NetApp recommends using an EC2 M5 Series instance for Oracle deployment, which is optimized for database workloads. The same instance type is also used to power a RDS instance for Oracle by AWS.

- Choose the correct vCPU and RAM combination based on workload characteristics.
- Add swap space to a VM. The default EC2 instance deployment does not create a swap space, which is not optimal for a database.

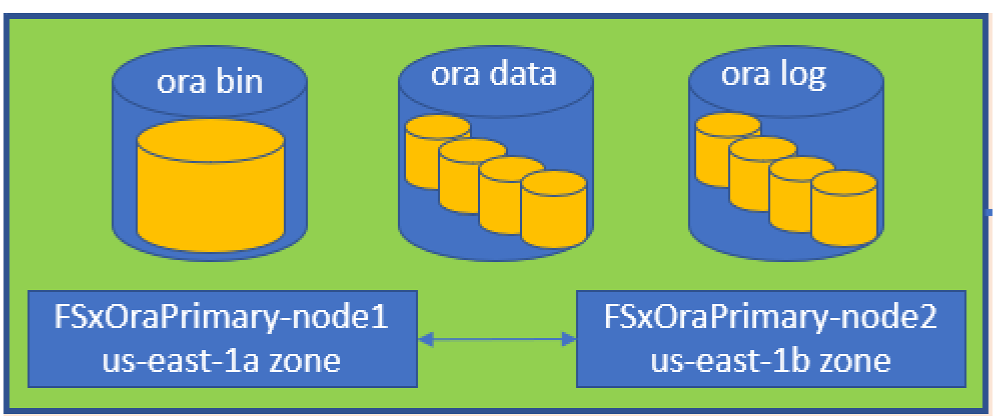
### Storage layout and settings

NetApp recommends the following storage layout:

- For NFS storage, the recommended volume layout is three volumes: one for the Oracle binary; one for Oracle data and a duplicate control file; and one for the Oracle active log, archived log, and control file.




- For iSCSI storage, the recommended volume layout is three volumes: one for the Oracle binary; one for Oracle data and a duplicate control file; and one for the Oracle active log, archived log, and control file. However, each data and log volume ideally should contain four LUNs. The LUNs are ideally balanced on the HA cluster nodes.




- For storage IOPS and throughput, you can choose the threshold for provisioned IOPS and throughput for the FSx storage cluster, and these parameters can be adjusted on the fly anytime the workload changes.
  - The auto IOPS setting is three IOPS per GiB of allocated storage capacity or user defined storage up to 80,000.

◦ The throughput level is incremented as follow: 128, 256, 512, 1024, 2045 MBps.

Review the Amazon FSx for NetApp ONTAP performance documentation when sizing throughput and IOPS.

#### NFS configuration

Linux, the most common operating system, includes native NFS capabilities. Oracle offers the direct NFS (dNFS) client natively integrated into Oracle. Oracle has supported NFSv3 for over 20 years. dNFS is supported with NFSv3 with all versions of Oracle. NFSv4 is supported with all OS's that follow the NFSv4 standard. dNFS support for NFSv4 requires Oracle 12.1.0.2 or higher. NFSv4.1 requires specific OS support. Consult the NetApp Interoperability Matrix Tool (IMT) for supported OS's. dNFS support for NFSv4.1 requires Oracle version 19.3.0.0 or higher.

Automated Oracle deployment using the NetApp automation toolkit automatically configures dNFS on NFSv3.

Other factors to consider:

- TCP slot tables are the NFS equivalent of host-bus-adapter (HBA) queue depth. These tables control the number of NFS operations that can be outstanding at any one time. The default value is usually 16, which is far too low for optimum performance. The opposite problem occurs on newer Linux kernels, which can automatically increase the TCP slot table limit to a level that saturates the NFS server with requests.

  For optimum performance and to prevent performance problems, adjust the kernel parameters that control the TCP slot tables to 128.

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

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

| File Type | Mount Options |
| --- | --- |
| • Control files<br>• Data files<br>• Redo logs | `rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536` |
| • ORACLE_HOME<br>• ORACLE_BASE | `rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536` |

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.

Next: Deployment procedures.

## Step-by-Step Oracle Deployment Procedures on AWS EC2 and FSx

Previous: Solution architecture.

### 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:

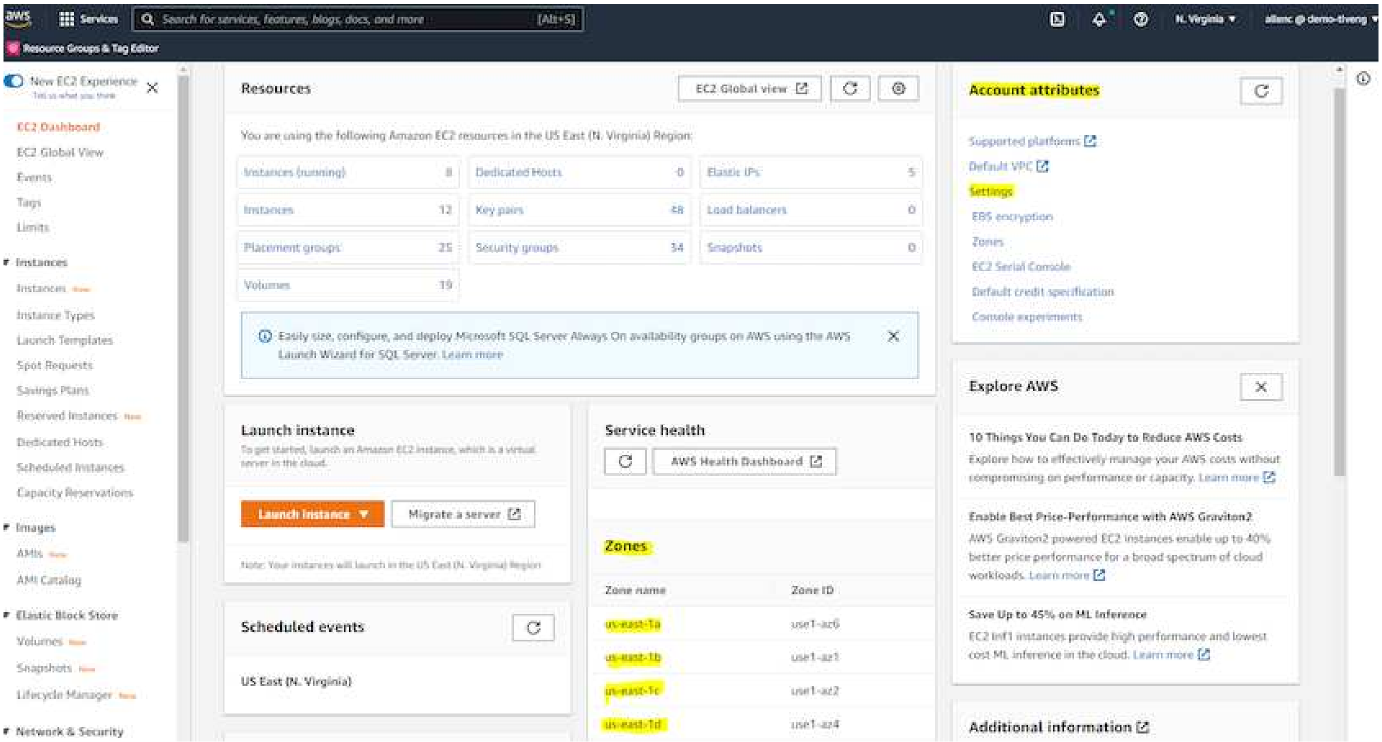
- Set up to use Amazon EC2

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**

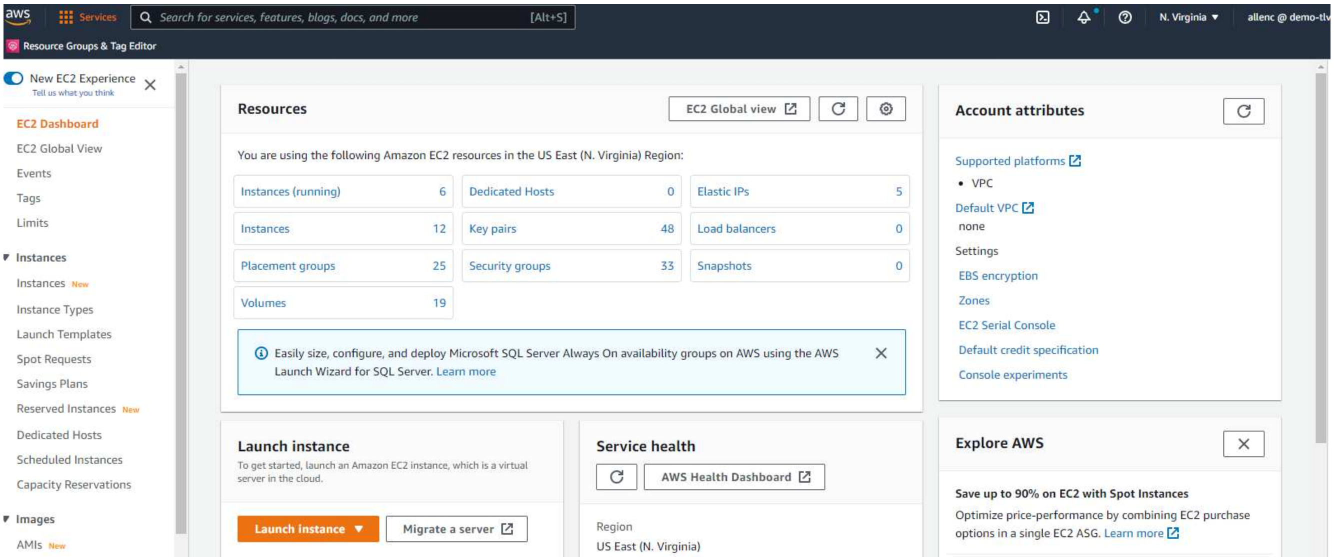
See the tutorial Get started with Amazon EC2 Linux instances for step-by-step deployment procedures and best practices.

Key topics:

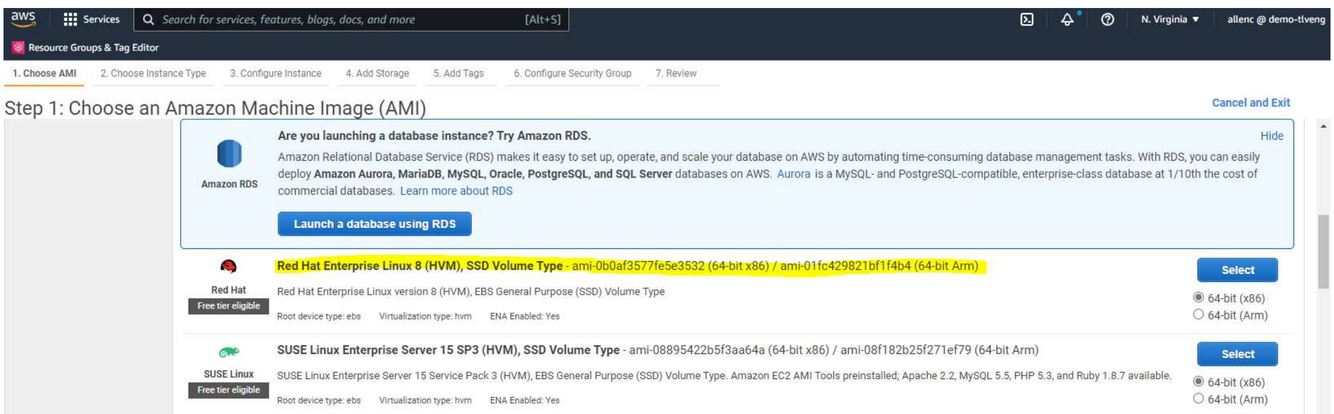
- Overview.
- Prerequisites.
- Step 1: Launch an instance.
- Step 2: Connect to your instance.
- Step 3: Clean up your instance.

The following screen shots demonstrate the deployment of an m5-type Linux instance with the EC2 console for running Oracle.

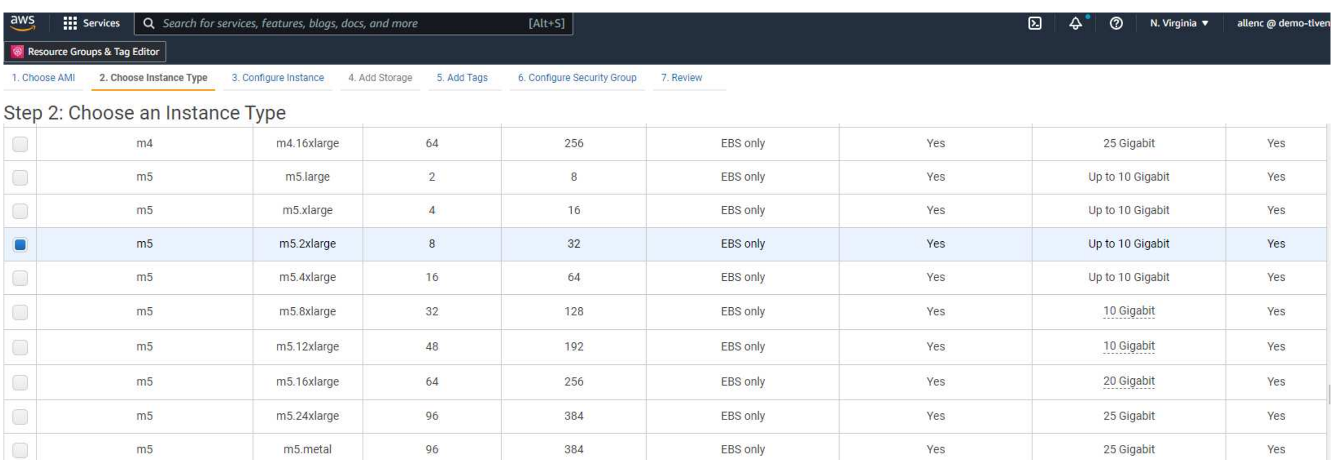
1. From the EC2 dashboard, click the yellow Launch Instance button to start the EC2 instance deployment workflow.




2. In Step 1, select "Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0b0af3577fe5e3532 (64-bit x86) / ami-01fc429821bf1f4b4 (64-bit Arm)."

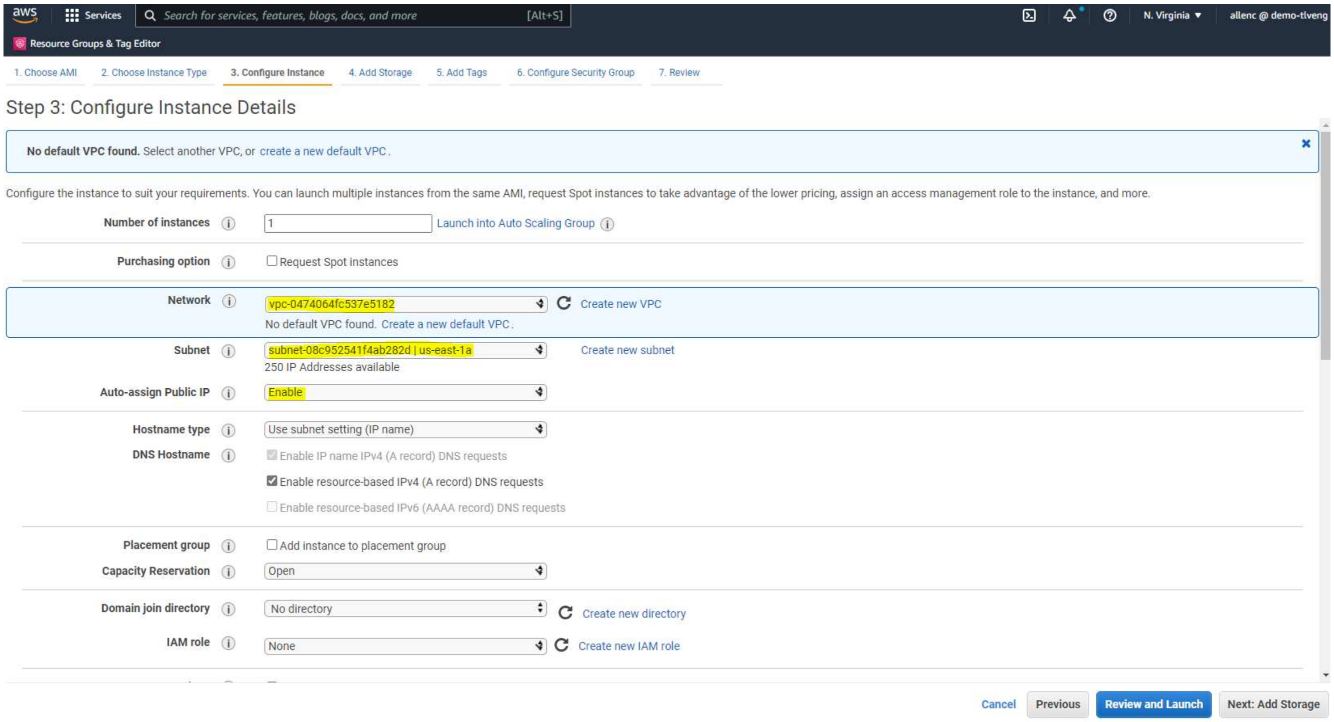



3. In Step 2, select an m5 instance type with the appropriate CPU and memory allocation based on your Oracle database workload. Click "Next: Configure Instance Details."

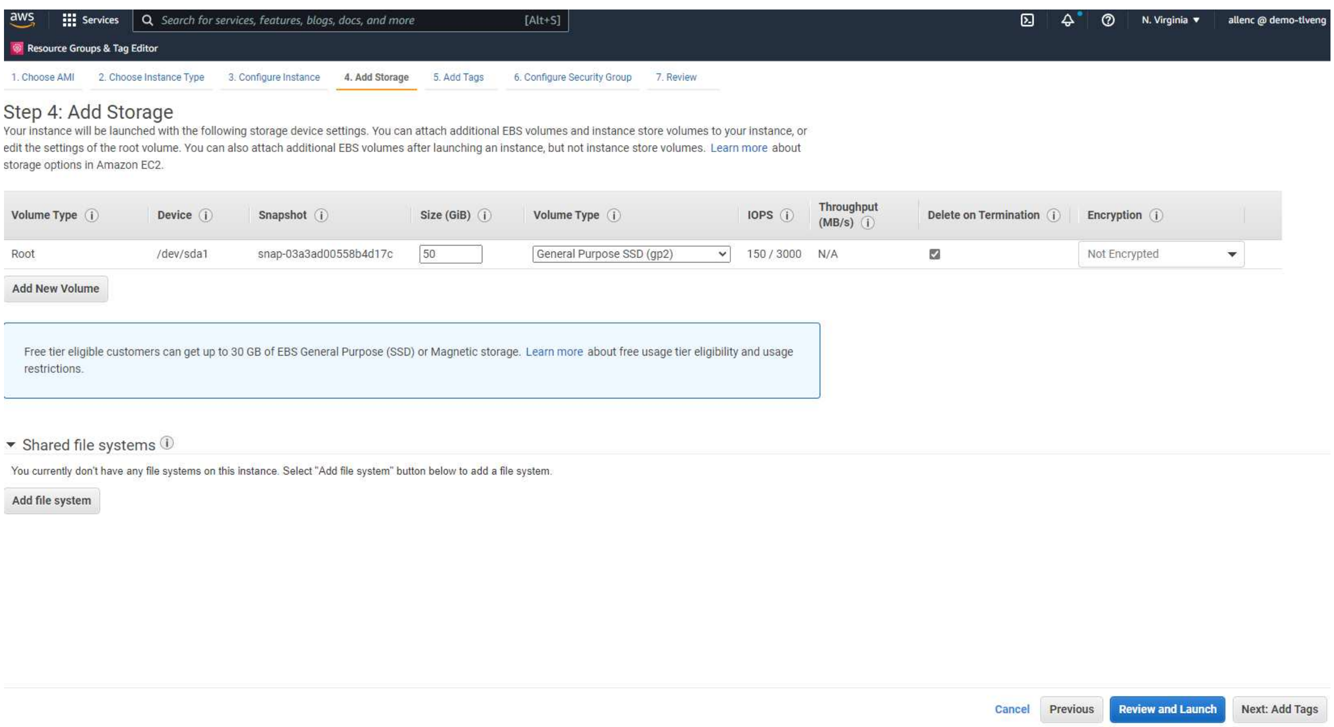



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| m4 | m4.16xlarge | 64 | 256 | EBS only | Yes | 25 Gigabit | Yes |
| m5 | m5.large | 2 | 8 | EBS only | Yes | Up to 10 Gigabit | Yes |
| m5 | m5.xlarge | 4 | 16 | EBS only | Yes | Up to 10 Gigabit | Yes |
| m5 | m5.2xlarge | 8 | 32 | EBS only | Yes | Up to 10 Gigabit | Yes |
| m5 | m5.4xlarge | 16 | 64 | EBS only | Yes | Up to 10 Gigabit | Yes |
| m5 | m5.8xlarge | 32 | 128 | EBS only | Yes | 10 Gigabit | Yes |
| m5 | m5.12xlarge | 48 | 192 | EBS only | Yes | 10 Gigabit | Yes |
| m5 | m5.16xlarge | 64 | 256 | EBS only | Yes | 20 Gigabit | Yes |
| m5 | m5.24xlarge | 96 | 384 | EBS only | Yes | 25 Gigabit | Yes |
| m5 | m5.metal | 96 | 384 | EBS only | Yes | 25 Gigabit | Yes |

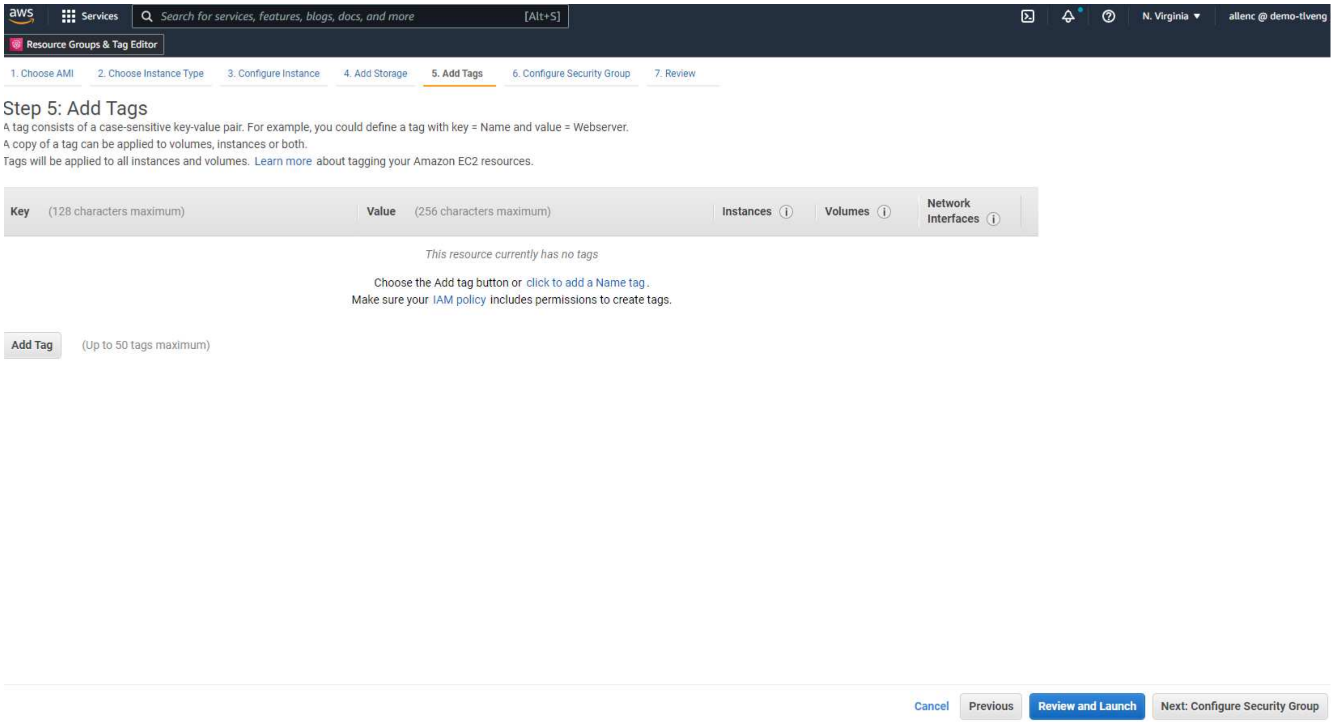
4. In Step 3, choose the VPC and subnet where the instance should be placed and enable public IP assignment. Click "Next: Add Storage."



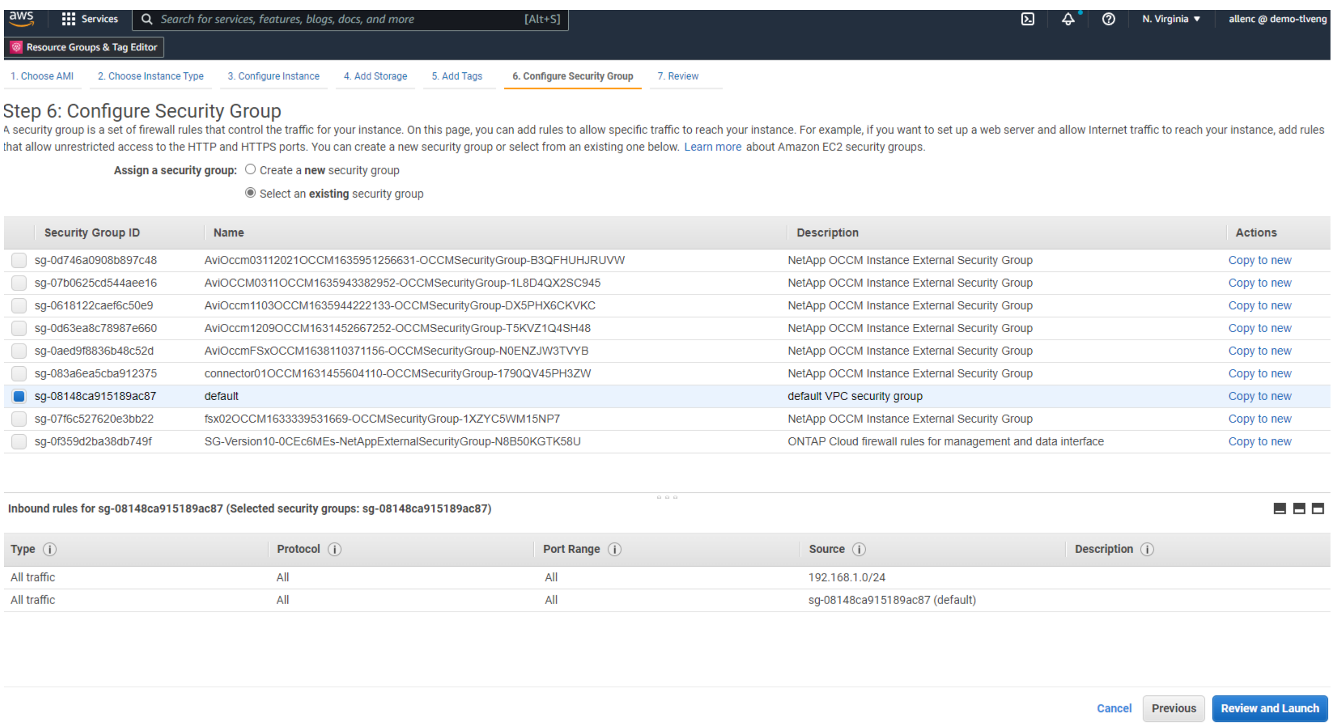

5. In Step 4, allocate enough space for the root disk. You may need the space to add a swap. By default, EC2 instance assign zero swap space, which is not optimal for running Oracle.



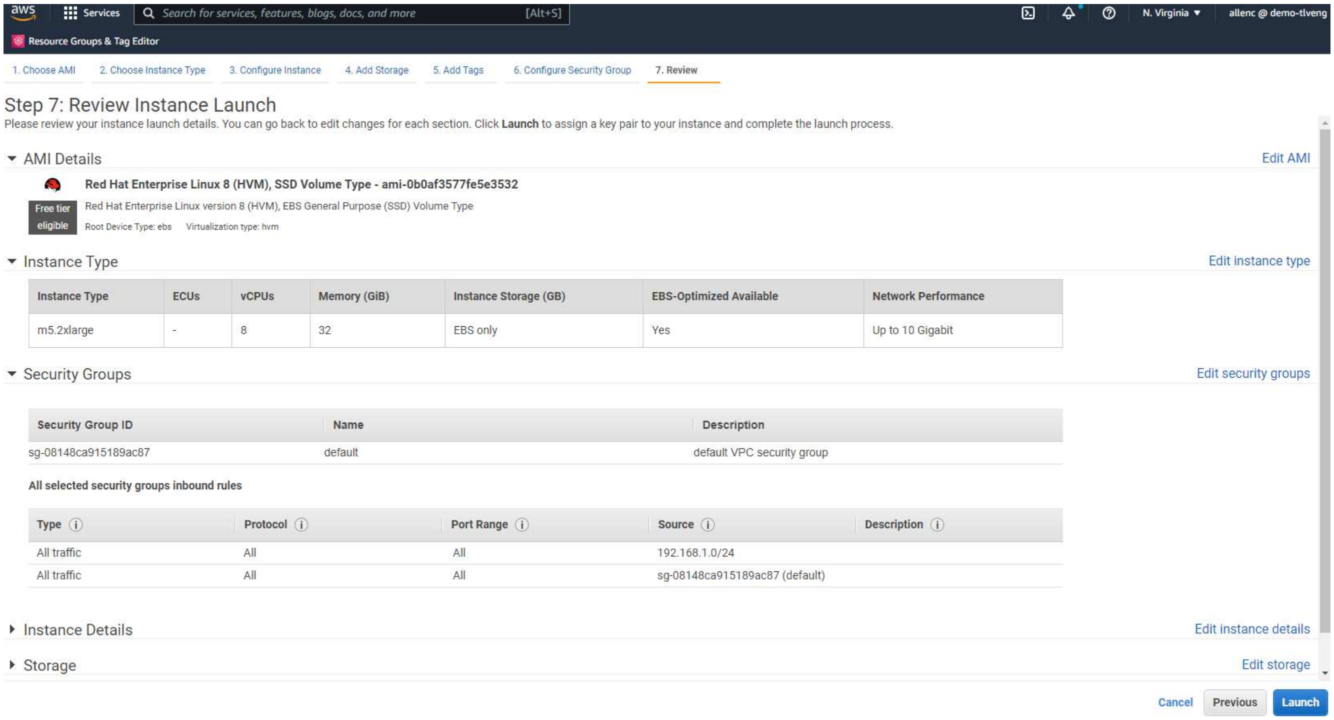

6. In Step 5, add a tag for instance identification if needed.

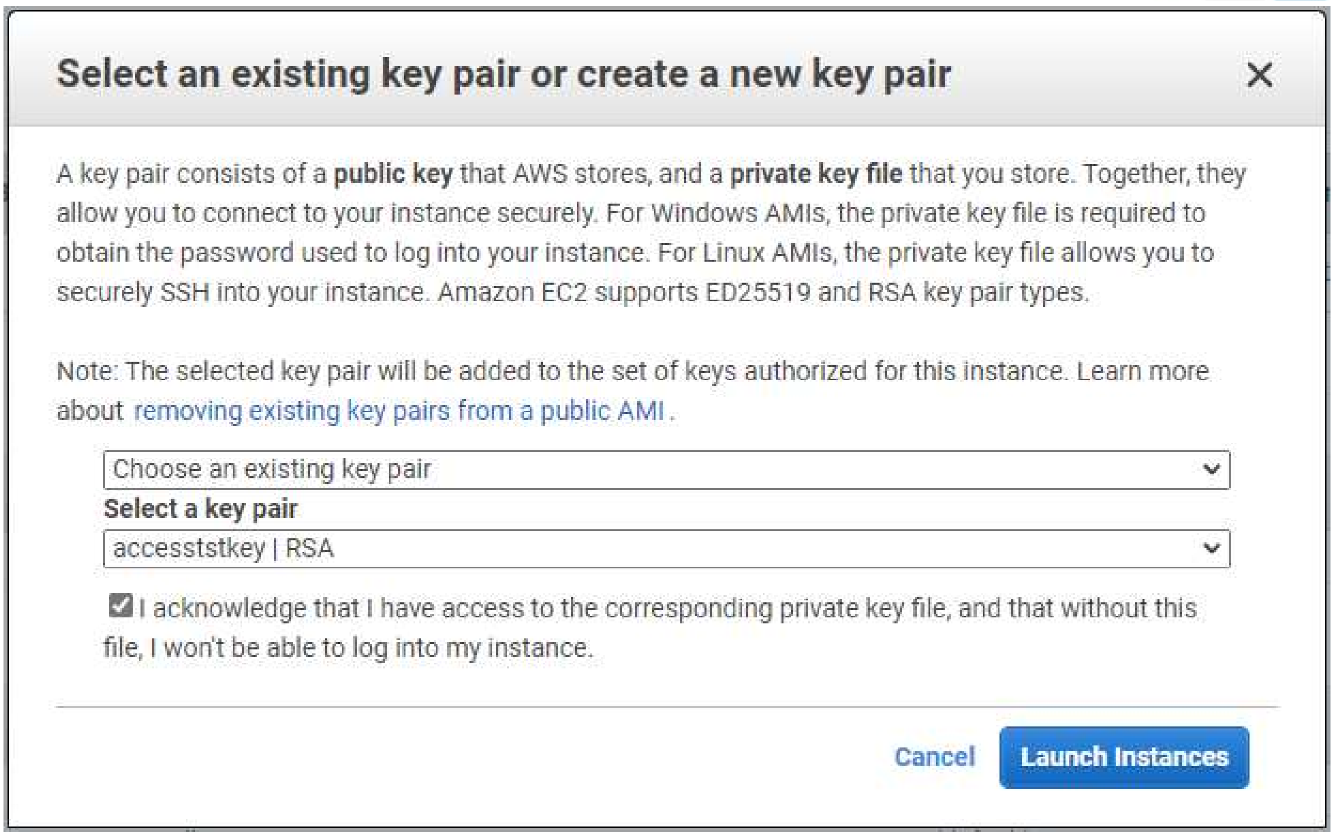



7. In Step 6, select an existing security group or create a new one with the desired inbound and outbound policy for the instance.




8. In Step 7, review the instance configuration summary, and click Launch to start instance deployment. You are prompted to create a key pair or select a key pair for access to the instance.




9. Log into EC2 instance using an SSH key pair. Make changes to your key name and instance IP address as appropriate.

```
ssh -i ora-db1v2.pem ec2-user@54.80.114.77
```

You need to create two EC2 instances as primary and standby Oracle servers in their designated availability zone as demonstrated in the architecture diagram.

### Provision FSx for ONTAP file systems for Oracle database storage

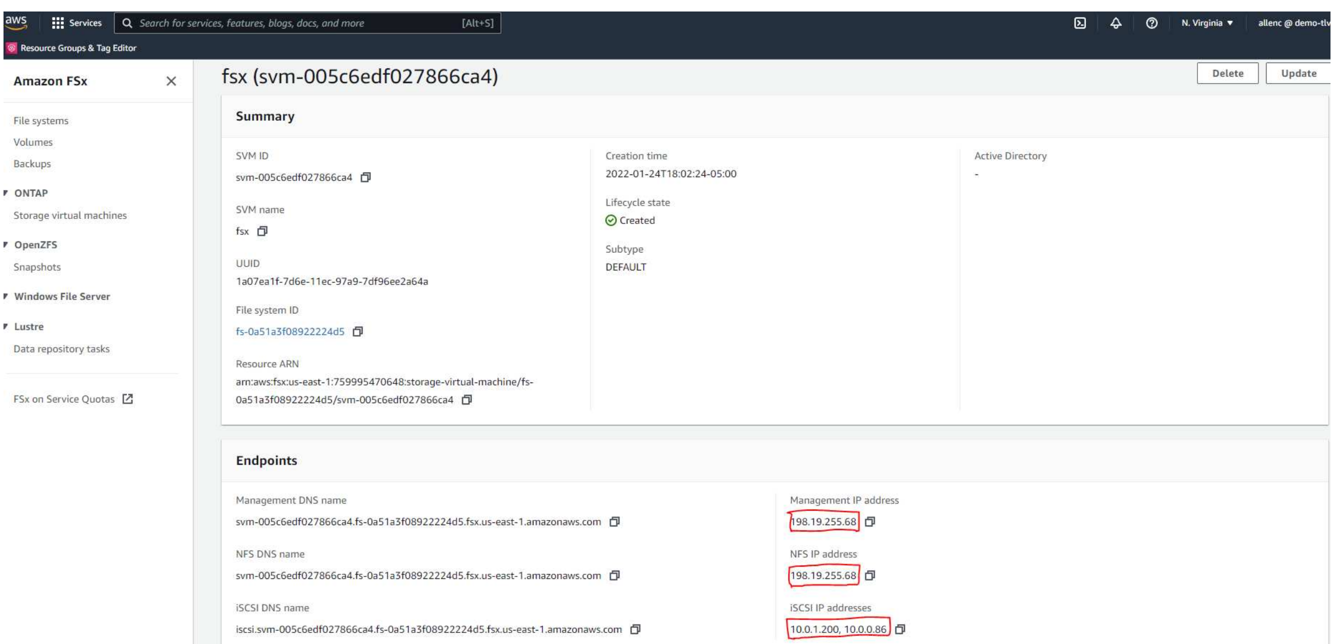
EC2 instance deployment allocates an EBS root volume for the OS. FSx for ONTAP file systems provides Oracle database storage volumes, including the Oracle binary, data, and log volumes. The FSx storage NFS volumes can be either provisioned from the AWS FSx console or from Oracle installation, and configuration automation that allocates the volumes as the user configures in a automation parameter file.

#### Creating FSx for ONTAP file systems

Referred to this documentation [Managing FSx for ONTAP file systems] for creating FSx for ONTAP file systems.

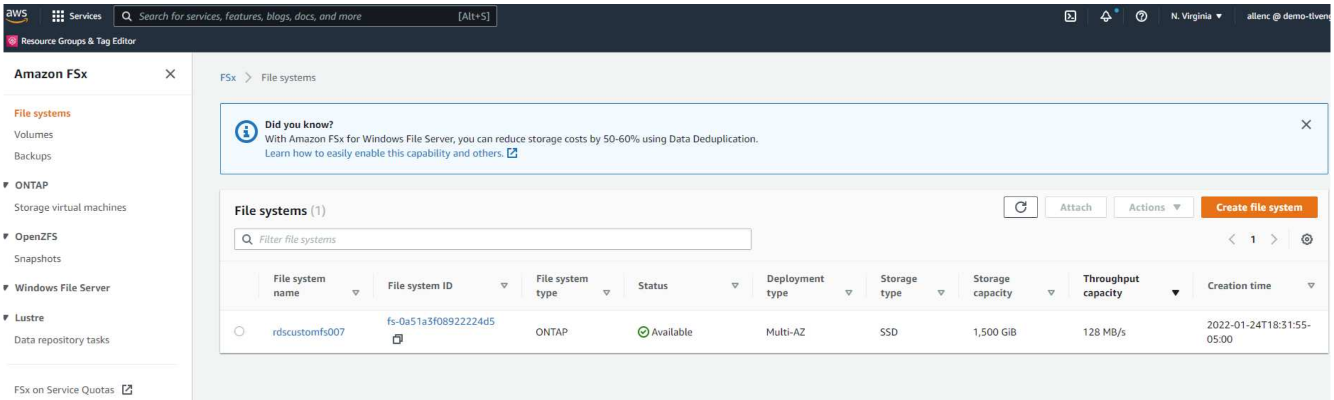
Key considerations:

- SSD storage capacity. Minimum 1024 GiB, maximum 192 TiB.
- Provisioned SSD IOPS. Based on workload requirements, a maximum of 80,000 SSD IOPS per file system.
- Throughput capacity.
- Set administrator fsxadmin/vsadmin password. Required for FSx configuration automation.
- Backup and maintenance. Disable automatic daily backups; database storage backup is executed through SnapCenter scheduling.
- Retrieve the SVM management IP address as well as protocol-specific access addresses from SVM details page. Required for FSx configuration automation.

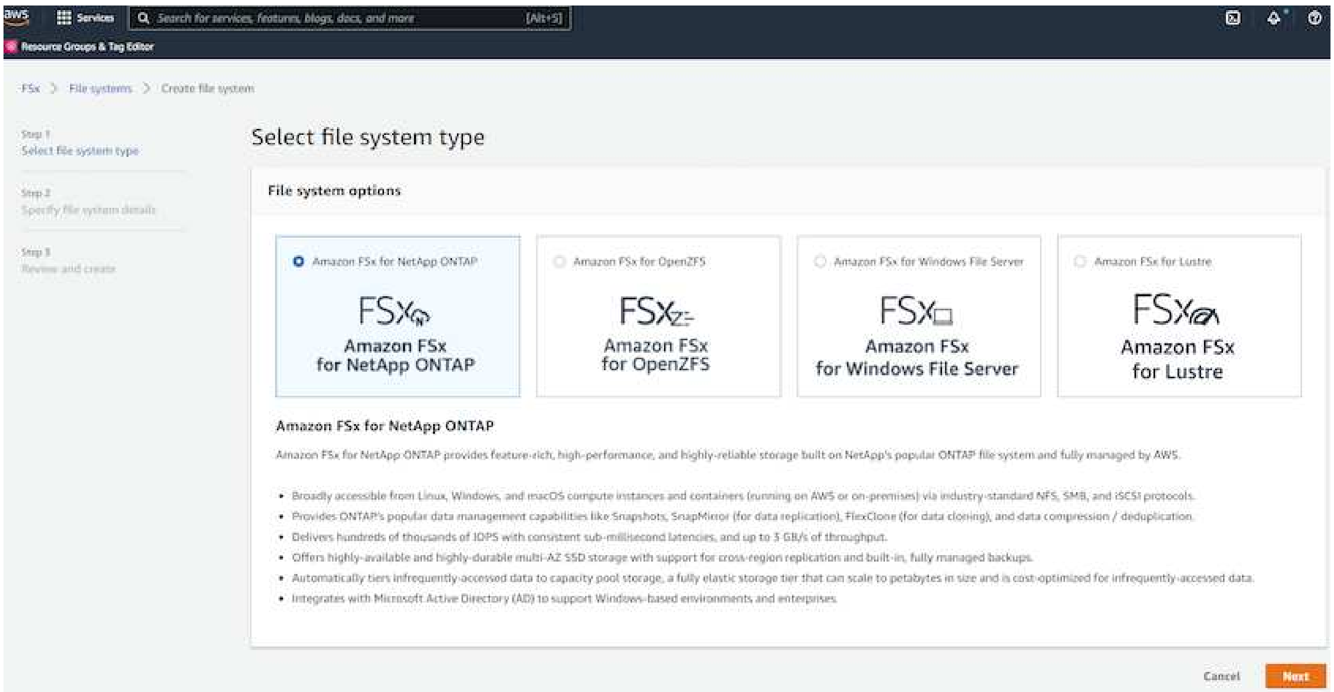



See the following step-by-step procedures for setting up either a primary or standby HA FSx cluster.

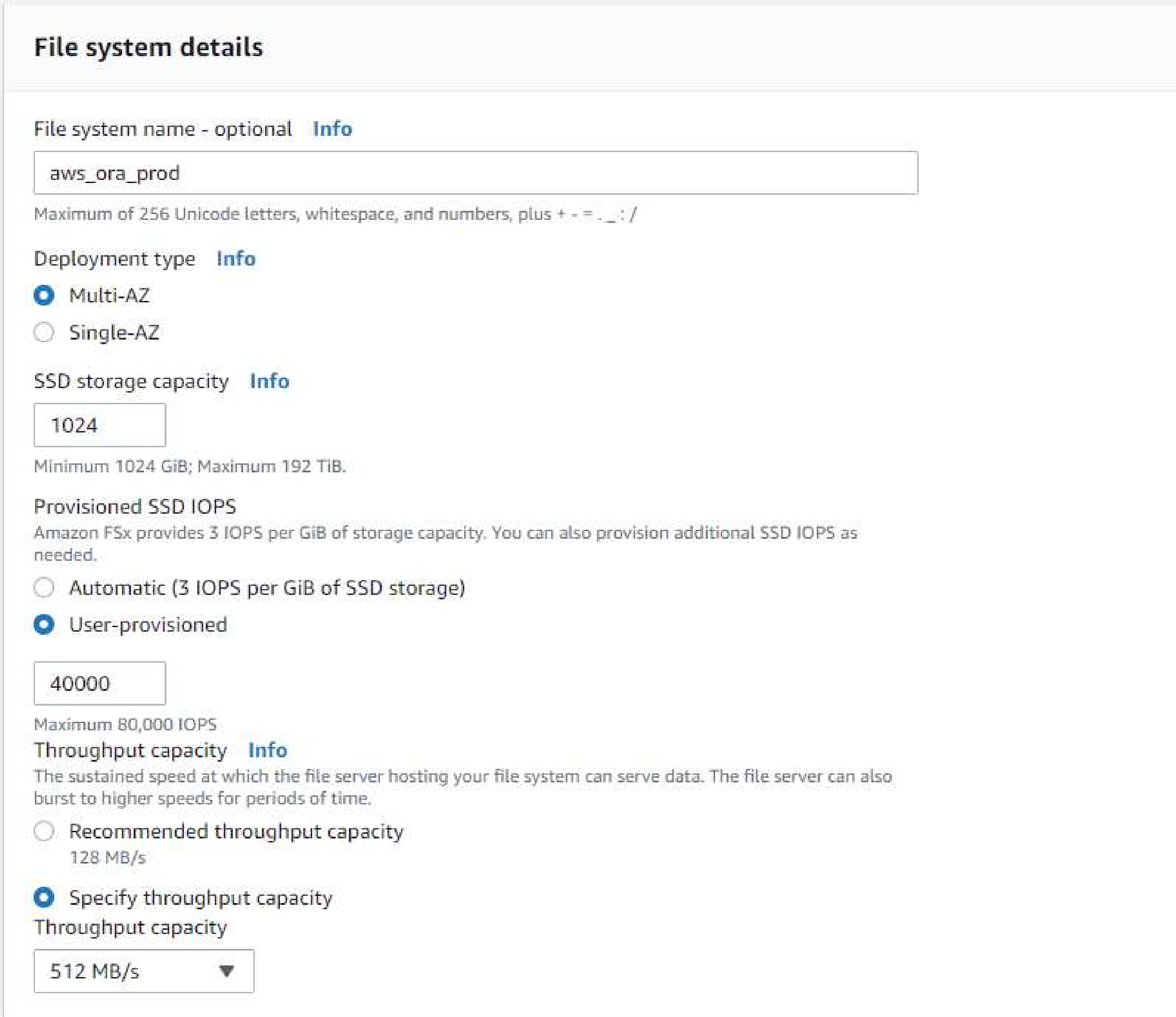
1. From the FSx console, click Create File System to start the FSx provision workflow.



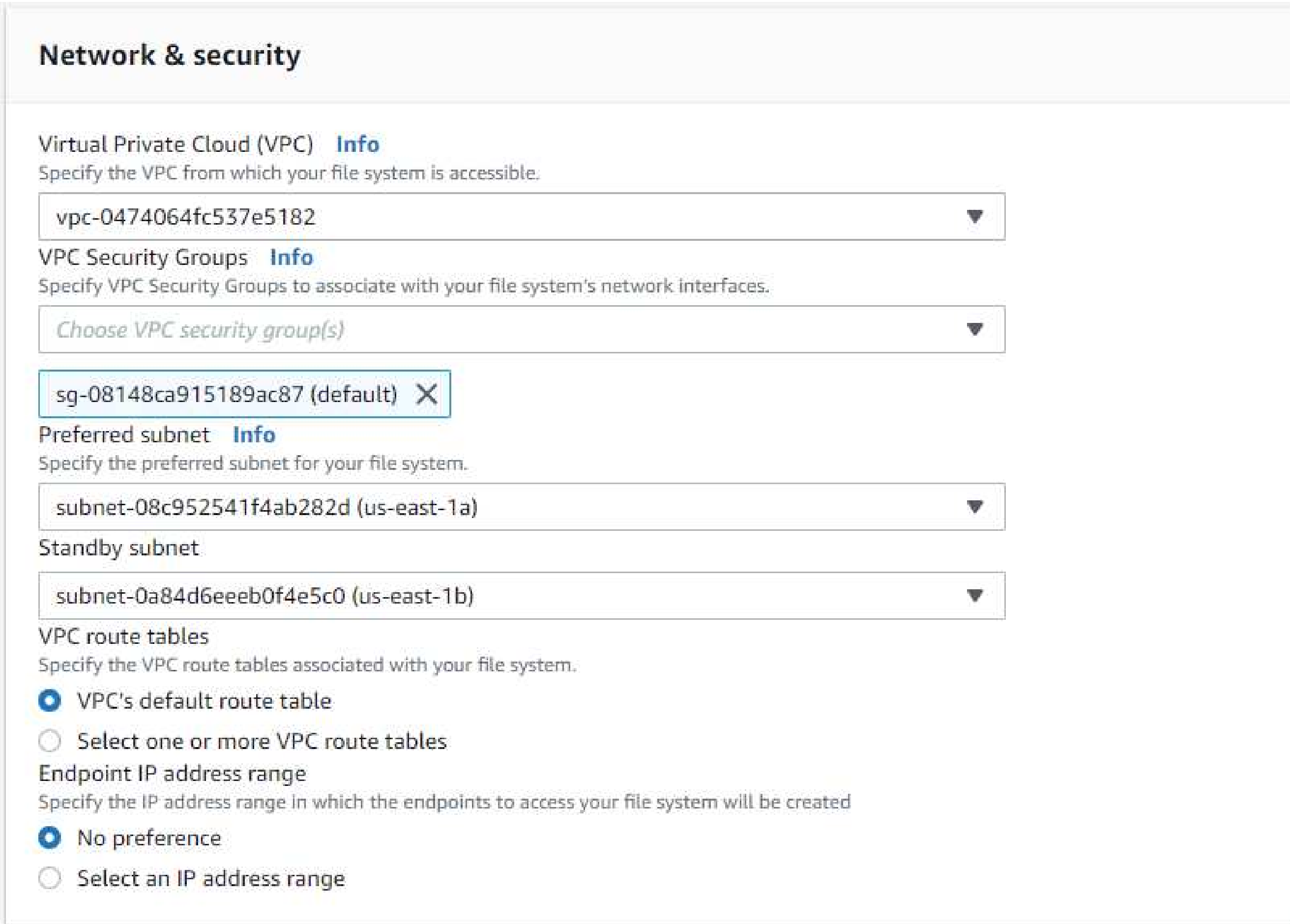

2. Select Amazon FSx for NetApp ONTAP. Then click Next.



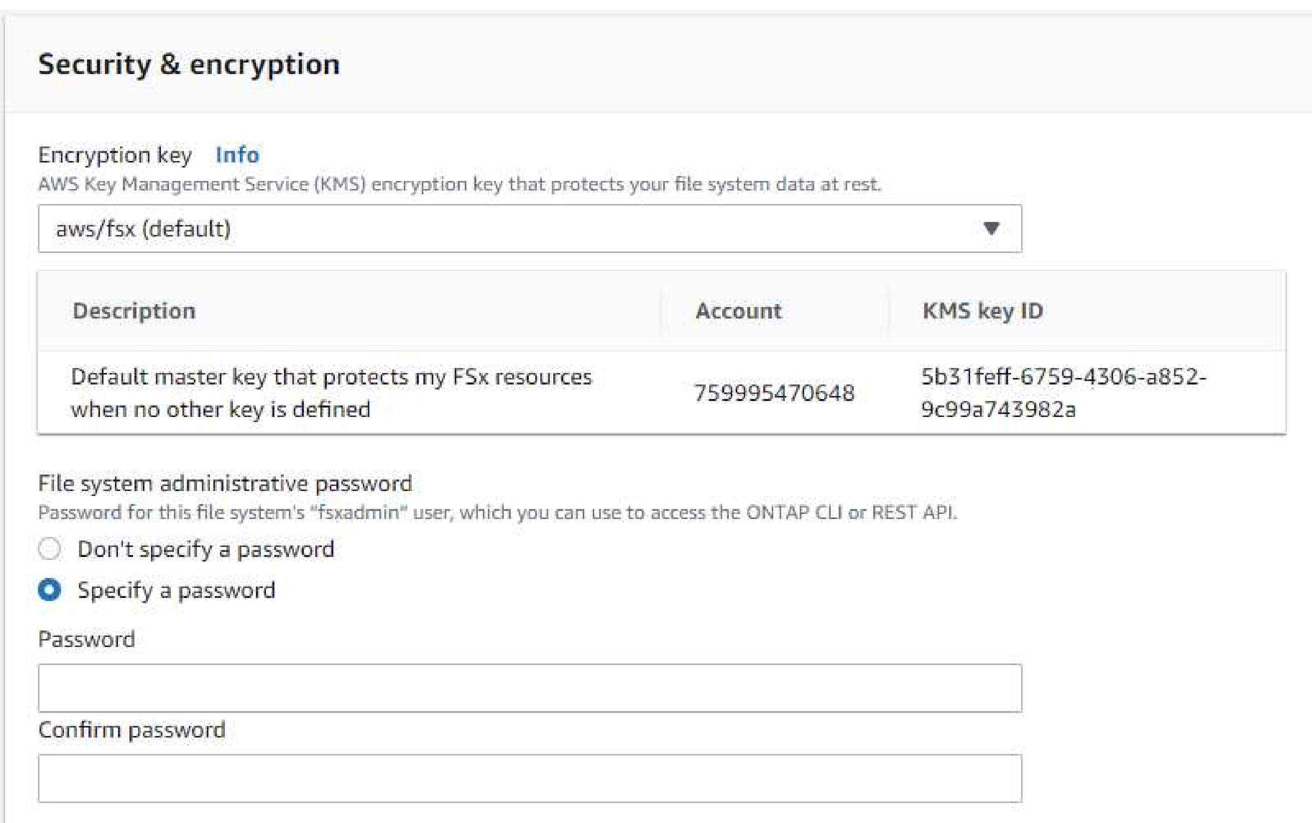

3. Select Standard Create and, in File System Details, name your file system, Multi-AZ HA. Based on your database workload, choose either Automatic or User-Provisioned IOPS up to 80,000 SSD IOPS. FSx storage comes with up to 2TiB NVMe caching at the backend that can deliver even higher measured IOPS.



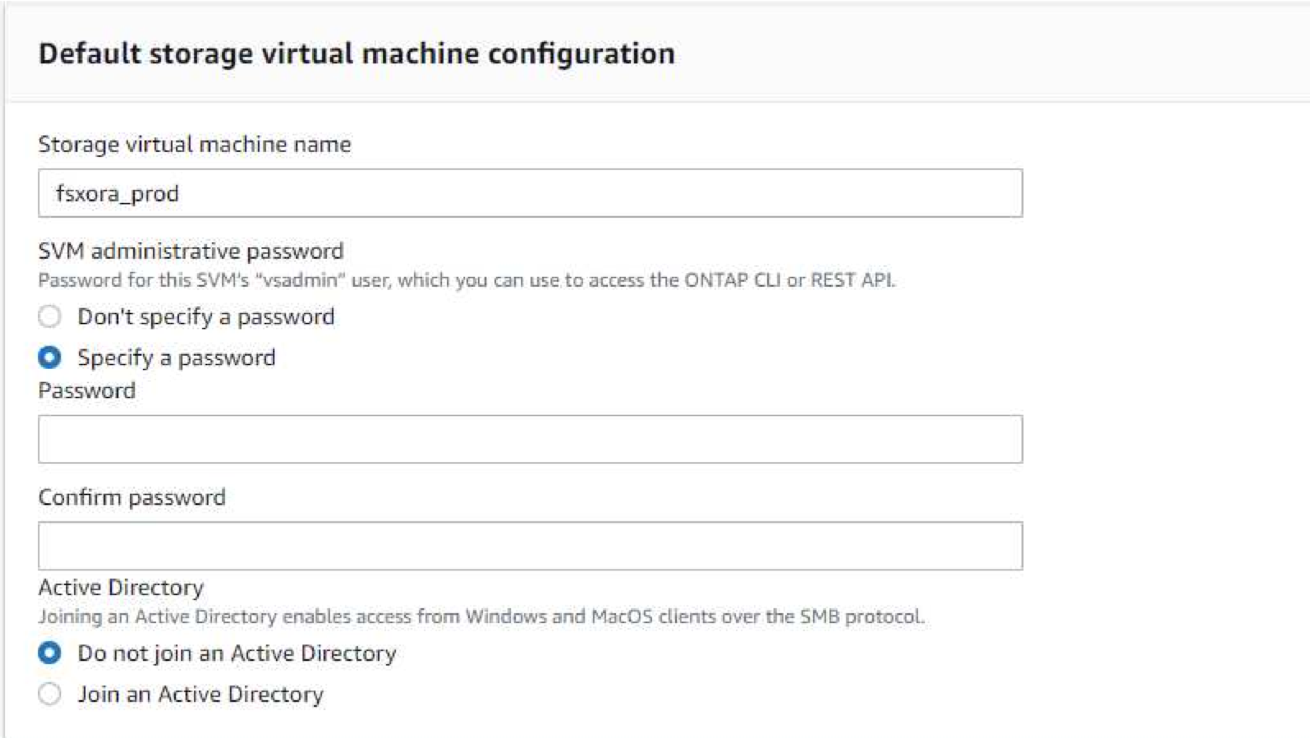

4. In the Network & Security section, select the VPC, security group, and subnets. These should be created before FSx deployment. Based on the role of the FSx cluster (primary or standby), place the FSx storage nodes in the appropriate zones.



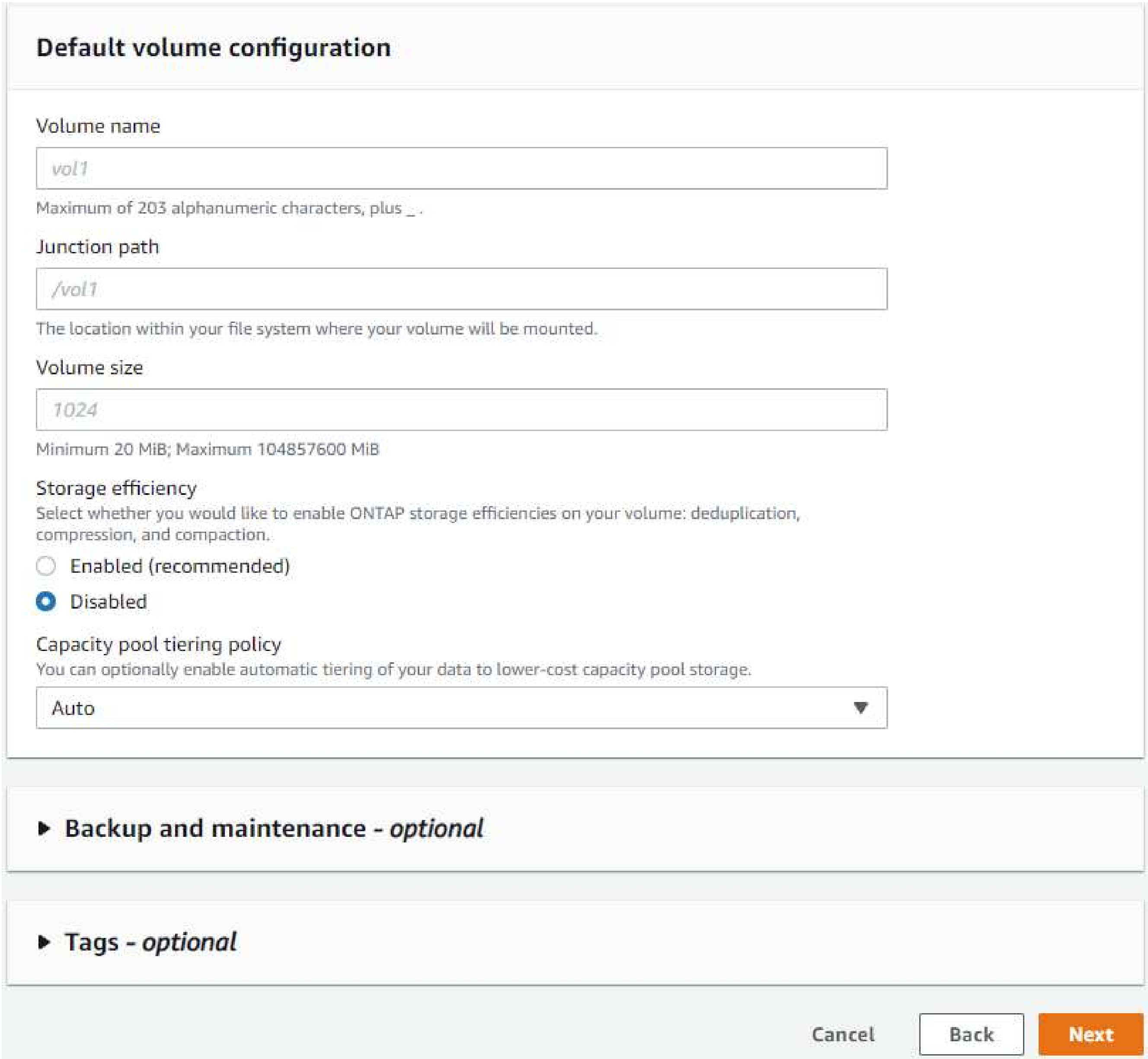

5. In the Security & Encryption section, accept the default, and enter the fsxadmin password.



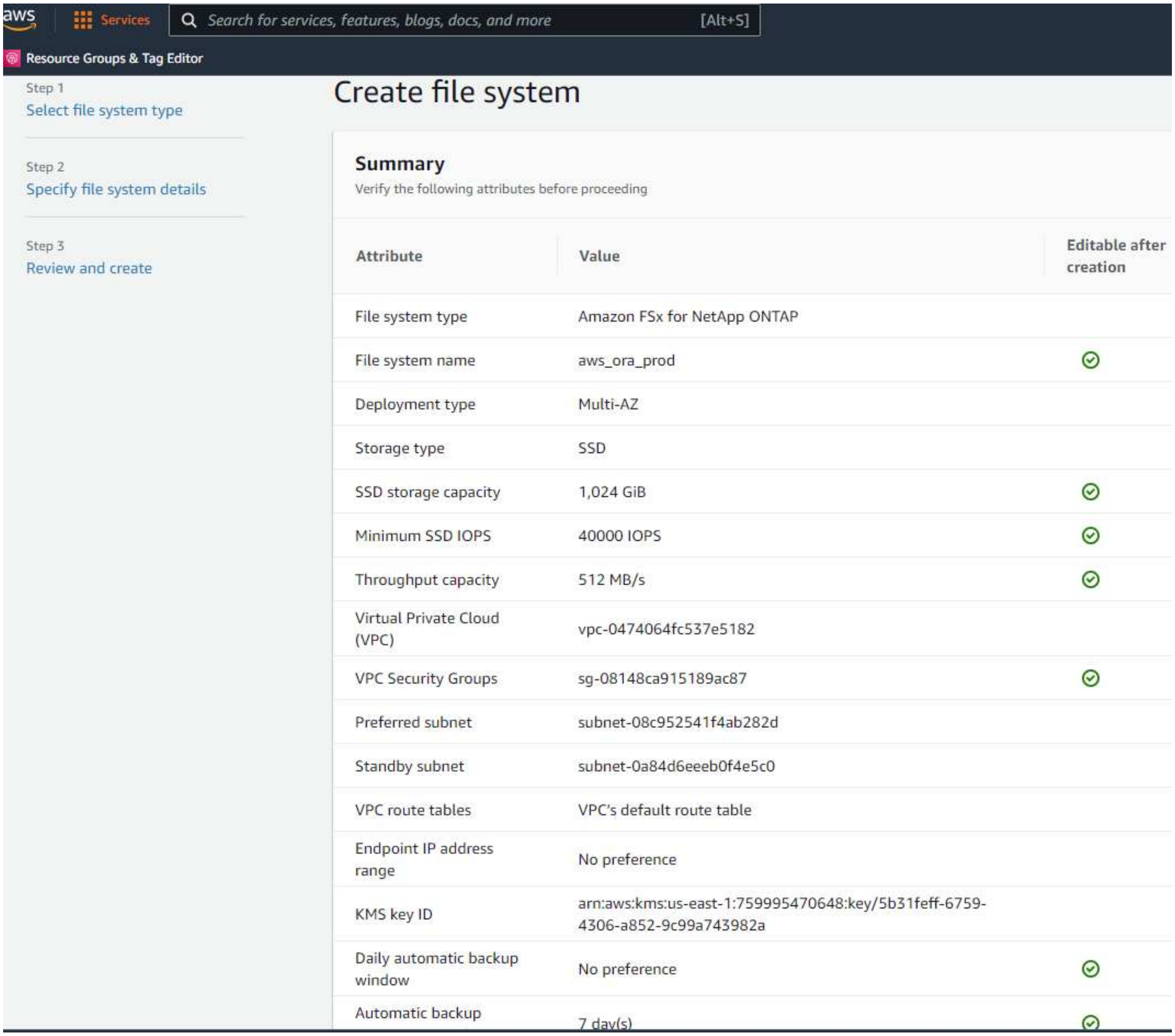
6. Enter the SVM name and the vsadmin password.




7. Leave the volume configuration blank; you do not need to create a volume at this point.




8. Review the Summary page, and click Create File System to complete FSx file system provision.




**Provisioning of database volumes for Oracle database**

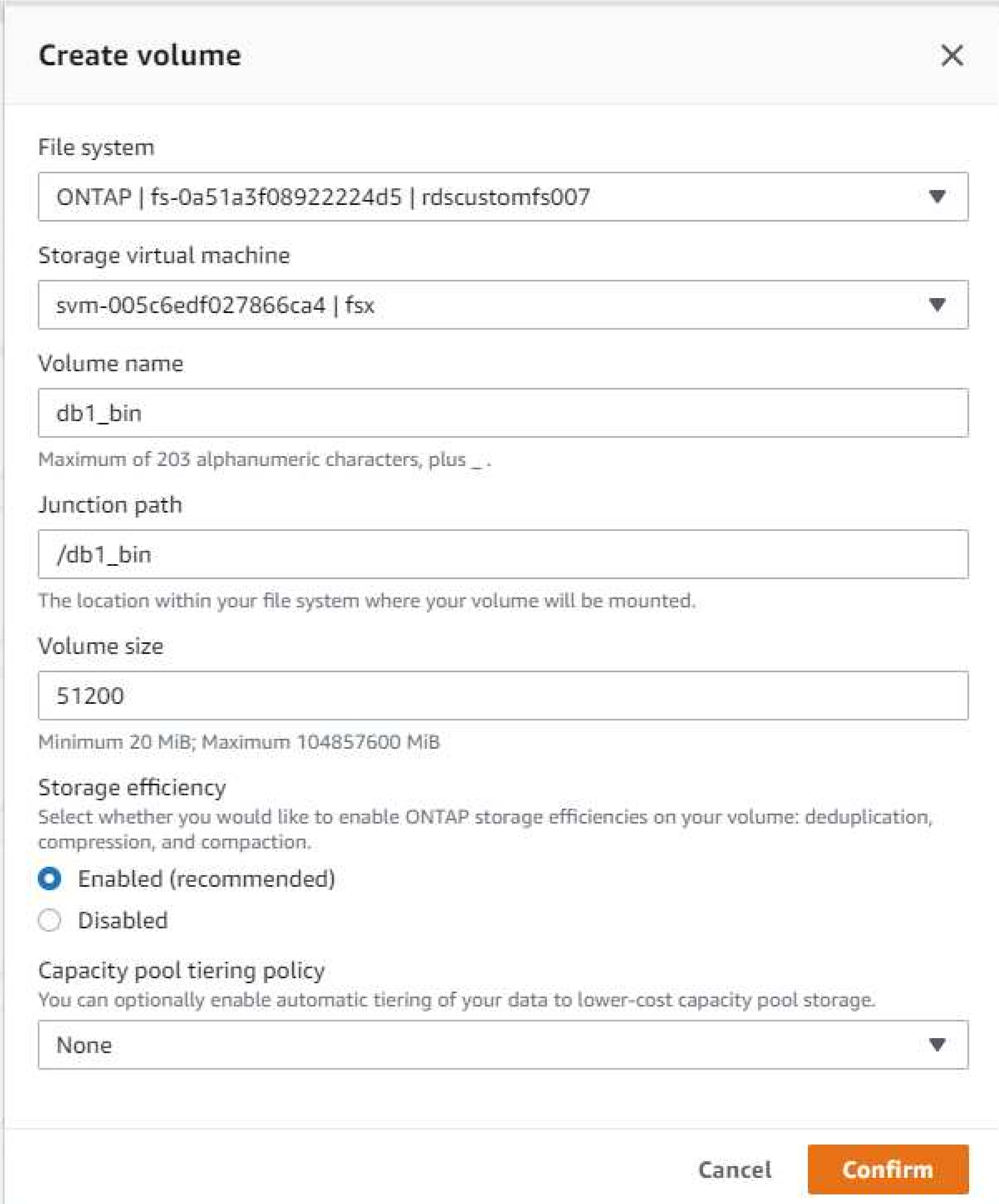
See Managing FSx for ONTAP volumes - creating a volume for details.

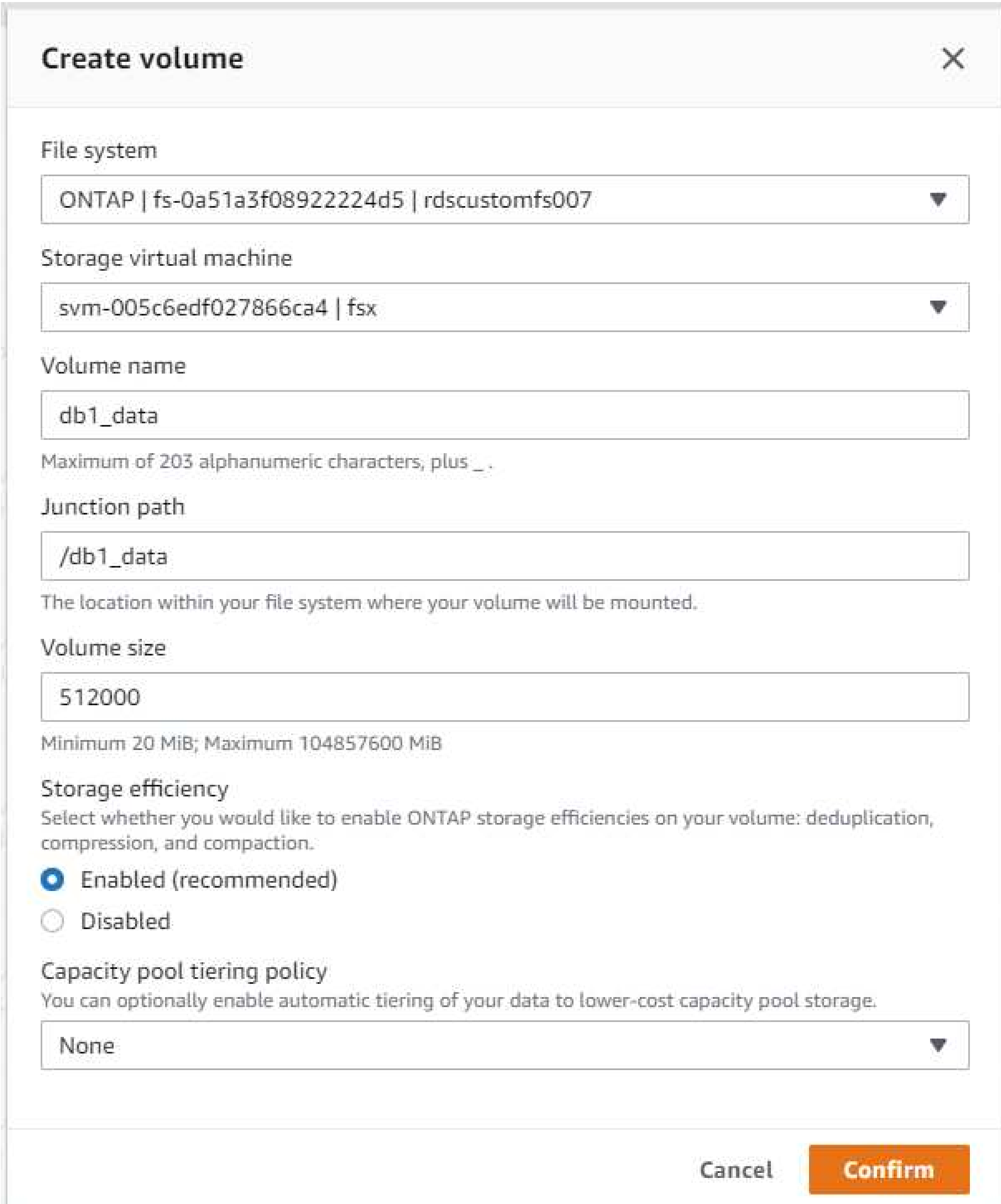
Key considerations:

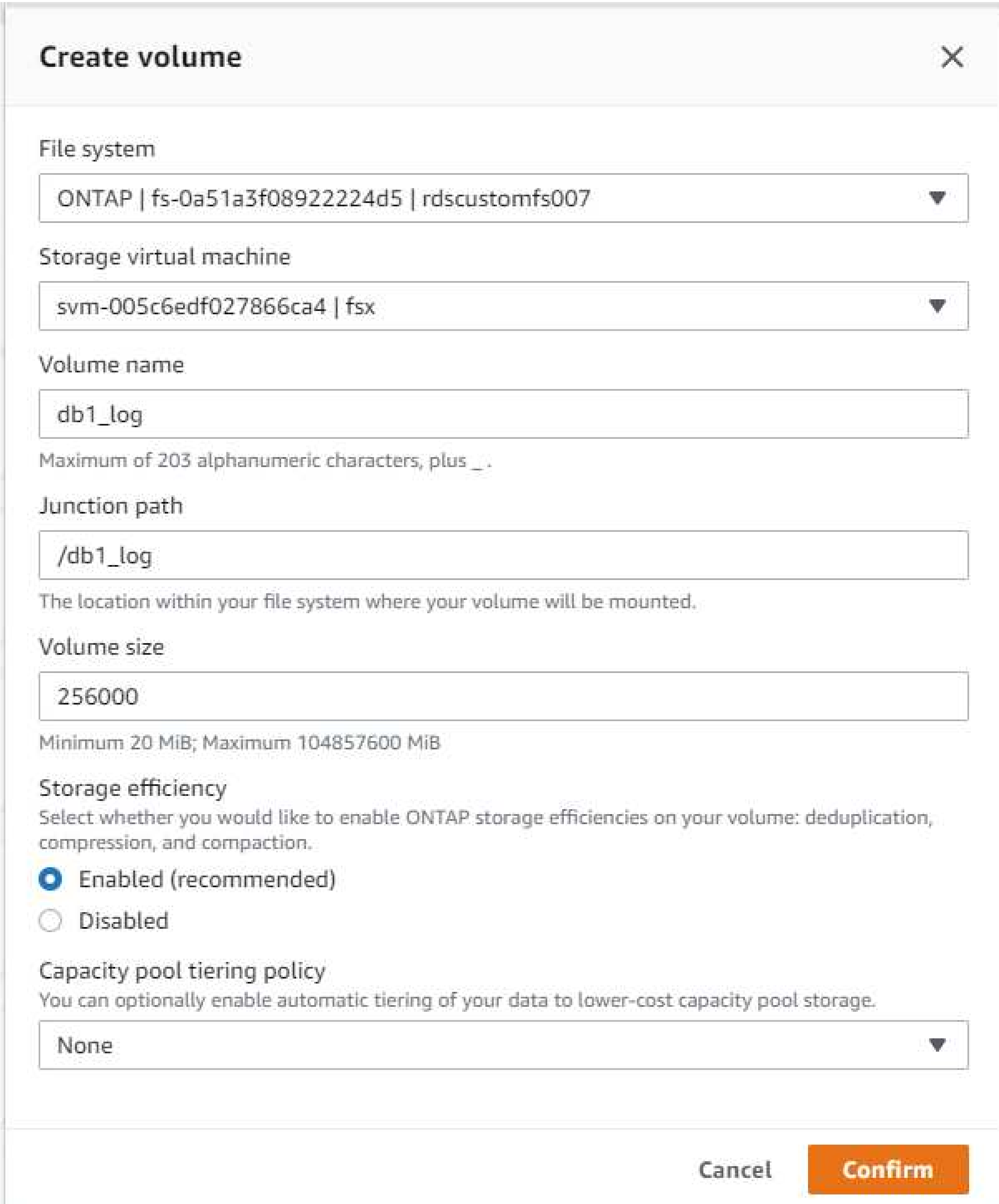
- Sizing the database volumes appropriately.
- Disabling capacity pool tiering policy for performance configuration.
- Enabling Oracle dNFS for NFS storage volumes.
- Setting up multipath for iSCSI storage volumes.

### Create database volume from FSx console

From the AWS FSx console, you can create three volumes for Oracle database file storage: one for the Oracle binary, one for the Oracle data, and one for the Oracle log. Make sure that volume naming matches the Oracle host name (defined in the hosts file in the automation toolkit) for proper identification. In this example, we use db1 as the EC2 Oracle host name instead of a typical IP-address-based host name for an EC2 instance.


Create volume
File system
ONTAP | fs-0a51a3f08922224d5 | rdscustomfs007
Storage virtual machine
svm-005c6edf027866ca4 | fsx
Volume name
db1_bin
Maximum of 203 alphanumeric characters, plus _ .
Junction path
/db1_bin
The location within your file system where your volume will be mounted.
Volume size
51200
Minimum 20 MiB; Maximum 104857600 MiB
Storage efficiency
Select whether you would like to enable ONTAP storage efficiencies on your volume: deduplication, compression, and compaction.
Enabled (recommended)
Disabled
Capacity pool tiering policy
You can optionally enable automatic tiering of your data to lower-cost capacity pool storage.
None
Cancel
Confirm


Create volume
File system
ONTAP | fs-0a51a3f08922224d5 | rdscustomfs007
Storage virtual machine
svm-005c6edf027866ca4 | fsx
Volume name
db1_data
Maximum of 203 alphanumeric characters, plus _ .
Junction path
/db1_data
The location within your file system where your volume will be mounted.
Volume size
512000
Minimum 20 MiB; Maximum 104857600 MiB
Storage efficiency
Select whether you would like to enable ONTAP storage efficiencies on your volume: deduplication, compression, and compaction.
Enabled (recommended)
Disabled
Capacity pool tiering policy
You can optionally enable automatic tiering of your data to lower-cost capacity pool storage.
None
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 `accessstkey.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 accessstkey.pem -e @vars/fsx_vars.yml -t requirements_config
   ```

   ```
   ansible-playbook -i hosts rds_preclone_config.yml -u ec2-user --private
   -key accessstkey.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`, `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.

```
FsxId00164454fac5591e6::> cluster peer show
Peer Cluster Name         Cluster Serial Number Availability   Authentication
------------------------- --------------------- -------------- --------------
FsxId0b6a95149d07aa82e    1-80-000011           Available      ok

FsxId00164454fac5591e6::> vserver peer show
            Peer        Peer                           Peering        Remote
Vserver     Vserver     State        Peer Cluster      Applications   Vserver
----------- ----------- ------------ ----------------- -------------- ---------
svm_FSxOraSource
            svm_FSxOraTarget
                        peered       FsxId0b6a95149d07aa82e
                                                       snapmirror     svm_FSxOraTarget

FsxId00164454fac5591e6::>
```

4. Create target NFS volumes at the standby FSx cluster for each source volume at the primary FSx cluster. Replace the volume name as appropriate for your environment.

```
vol create -volume dr_db1_bin -aggregate aggr1 -size 50G -state online
-policy default -type DP
```

```
vol create -volume dr_db1_data -aggregate aggr1 -size 500G -state online
-policy default -type DP
```

```
vol create -volume dr_db1_log -aggregate aggr1 -size 250G -state online
-policy default -type DP
```

5. You can also create iSCSI volumes and LUNs for the Oracle binary, Oracle data, and the Oracle log if the iSCSI protocol is employed for data access. Leave approximately 10% free space in the volumes for snapshots.

```
vol create -volume dr_db1_bin -aggregate aggr1 -size 50G -state online
-policy default -unix-permissions ---rwxr-xr-x -type RW
```

```
lun create -path /vol/dr_db1_bin/dr_db1_bin_01 -size 45G -ostype linux
```

```
vol create -volume dr_db1_data -aggregate aggr1 -size 500G -state online
-policy default -unix-permissions ---rwxr-xr-x -type RW
```

```
lun create -path /vol/dr_db1_data/dr_db1_data_01 -size 100G -ostype
linux
```

```
lun create -path /vol/dr_db1_data/dr_db1_data_02 -size 100G -ostype
linux
```

```
lun create -path /vol/dr_db1_data/dr_db1_data_03 -size 100G -ostype
linux
```

```
lun create -path /vol/dr_db1_data/dr_db1_data_04 -size 100G -ostype
linux
```

vol create -volume dr_db1_log -aggregate aggr1 -size 250G -state online -policy default -unix-permissions ---rwxr-xr-x -type RW

```
lun create -path /vol/dr_db1_log/dr_db1_log_01 -size 45G -ostype linux
```

```
lun create -path /vol/dr_db1_log/dr_db1_log_02 -size 45G -ostype linux
```

```
lun create -path /vol/dr_db1_log/dr_db1_log_03 -size 45G -ostype linux
```

```
lun create -path /vol/dr_db1_log/dr_db1_log_04 -size 45G -ostype linux
```

6. For iSCSI LUNs, create mapping for the Oracle host initiator for each LUN, using the binary LUN as an example. Replace the igroup with an appropriate name for your environment, and increment the lun-id for each additional LUN.

```
lun mapping create -path /vol/dr_db1_bin/dr_db1_bin_01 -igroup ip-10-0-
1-136 -lun-id 0
```

```
lun mapping create -path /vol/dr_db1_data/dr_db1_data_01 -igroup ip-10-
0-1-136 -lun-id 1
```

7. Create a SnapMirror relationship between the primary and standby database volumes. Replace the appropriate SVM name for your environment.s

```
snapmirror create -source-path svm_FSxOraSource:db1_bin -destination
-path svm_FSxOraTarget:dr_db1_bin -vserver svm_FSxOraTarget -throttle
unlimited -identity-preserve false -policy MirrorAllSnapshots -type DP
```

```
snapmirror create -source-path svm_FSxOraSource:db1_data -destination
-path svm_FSxOraTarget:dr_db1_data -vserver svm_FSxOraTarget -throttle
unlimited -identity-preserve false -policy MirrorAllSnapshots -type DP
```

```
snapmirror create -source-path svm_FSxOraSource:db1_log -destination
-path svm_FSxOraTarget:dr_db1_log -vserver svm_FSxOraTarget -throttle
unlimited -identity-preserve false -policy MirrorAllSnapshots -type DP
```

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.

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

Read the README instructions carefully before attempting setup and failover testing.

> Replicating the Oracle binary from the primary to a standby cluster might have Oracle license implications. Contact your Oracle license representative for clarification. The alternative is to have Oracle installed and configured at the time of recovery and failover.

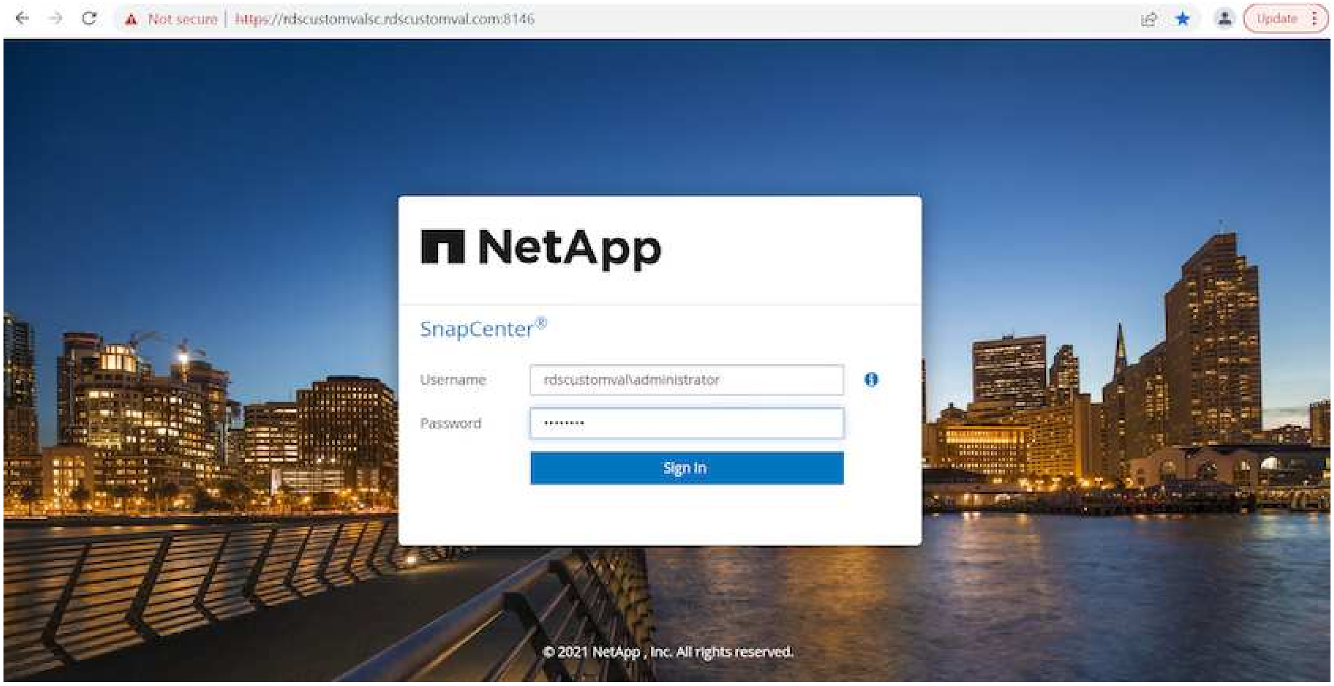
### SnapCenter Deployment

#### SnapCenter installation

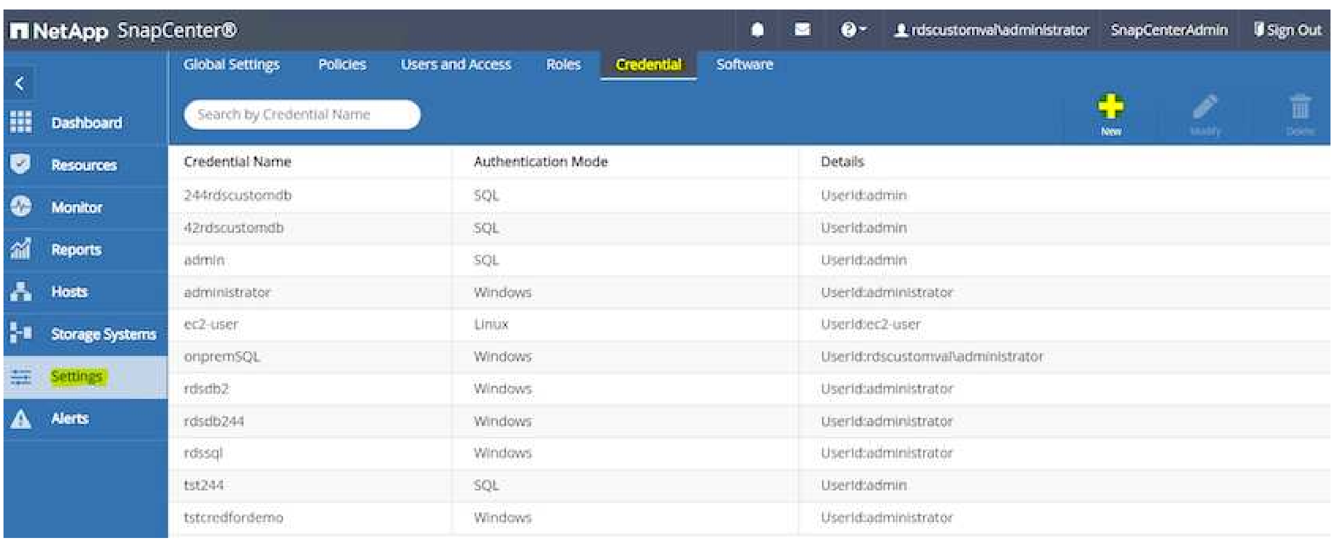
Follow Installing the SnapCenter Server to install SnapCenter server. This documentation covers how to install a standalone SnapCenter server. A SaaS version of SnapCenter is in beta review and could be available shortly. Check with your NetApp representative for availability if needed.

#### Configure SnapCenter plugin for EC2 Oracle host

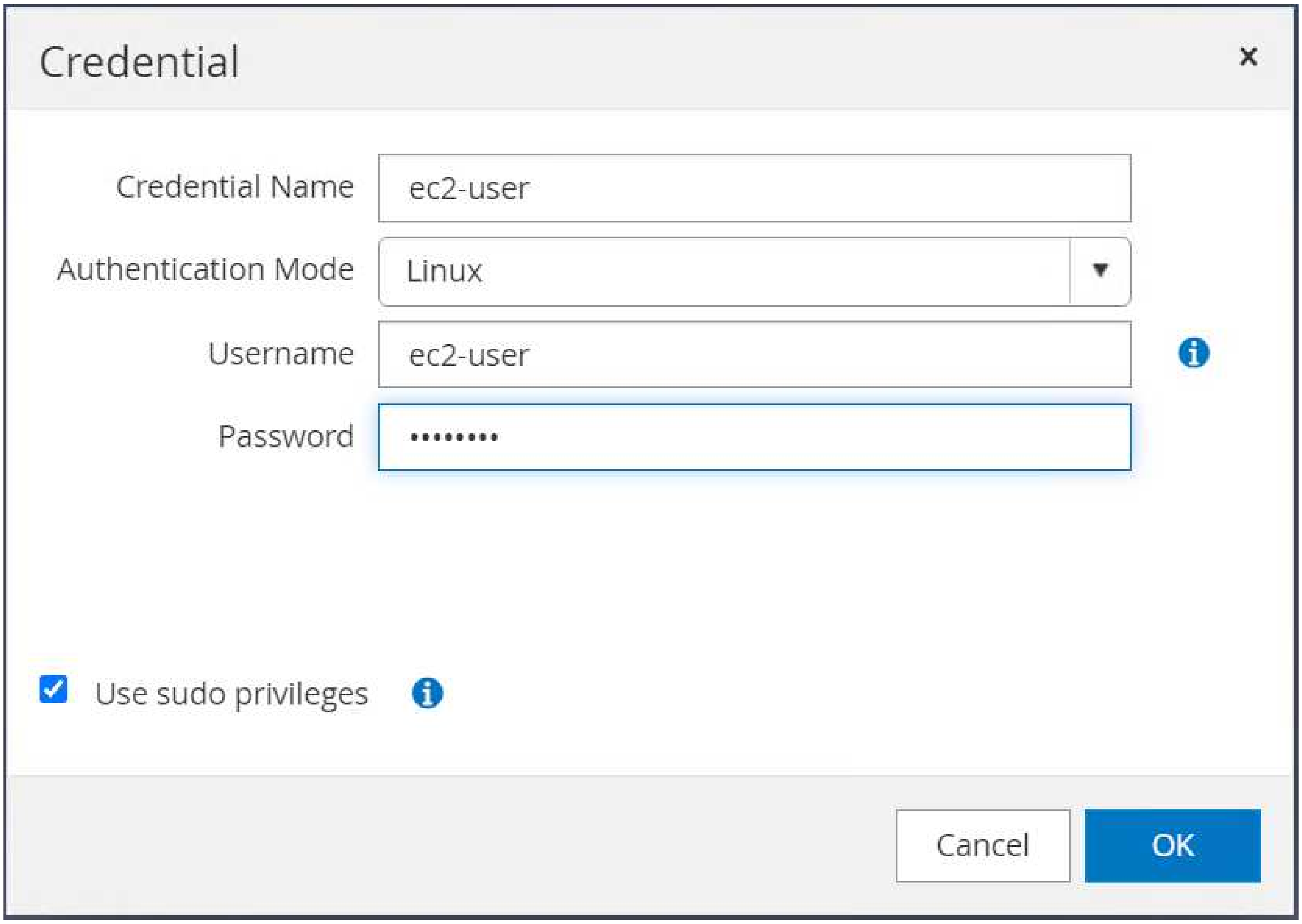
1. After automated SnapCenter installation, log into SnapCenter as an administrative user for the Window host on which the SnapCenter server is installed.




2. From the left-side menu, click Settings, and then Credential and New to add ec2-user credentials for SnapCenter plugin installation.




3. Reset the ec2-user password and enable password SSH authentication by editing the `/etc/ssh/sshd_config` file on the EC2 instance host.
4. Verify that the "Use sudo privileges" checkbox is selected. You just reset the ec2-user password in the previous step.



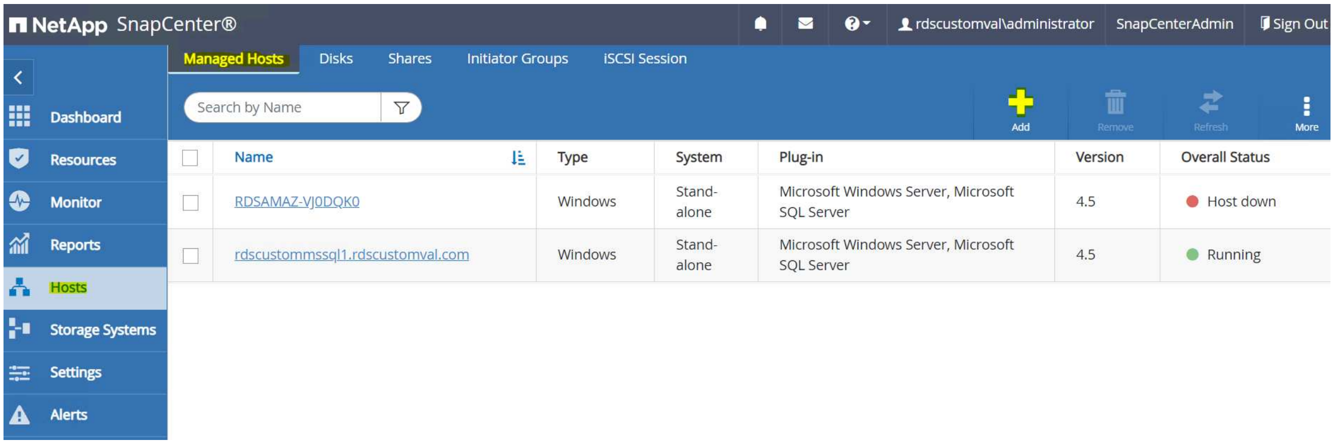

5. Add the SnapCenter server name and the IP address to the EC2 instance host file for name resolution.

```
[ec2-user@ip-10-0-0-151 ~]$ sudo vi /etc/hosts
[ec2-user@ip-10-0-0-151 ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
10.0.1.233  rdscustomvalsc.rdscustomval.com rdscustomvalsc
```

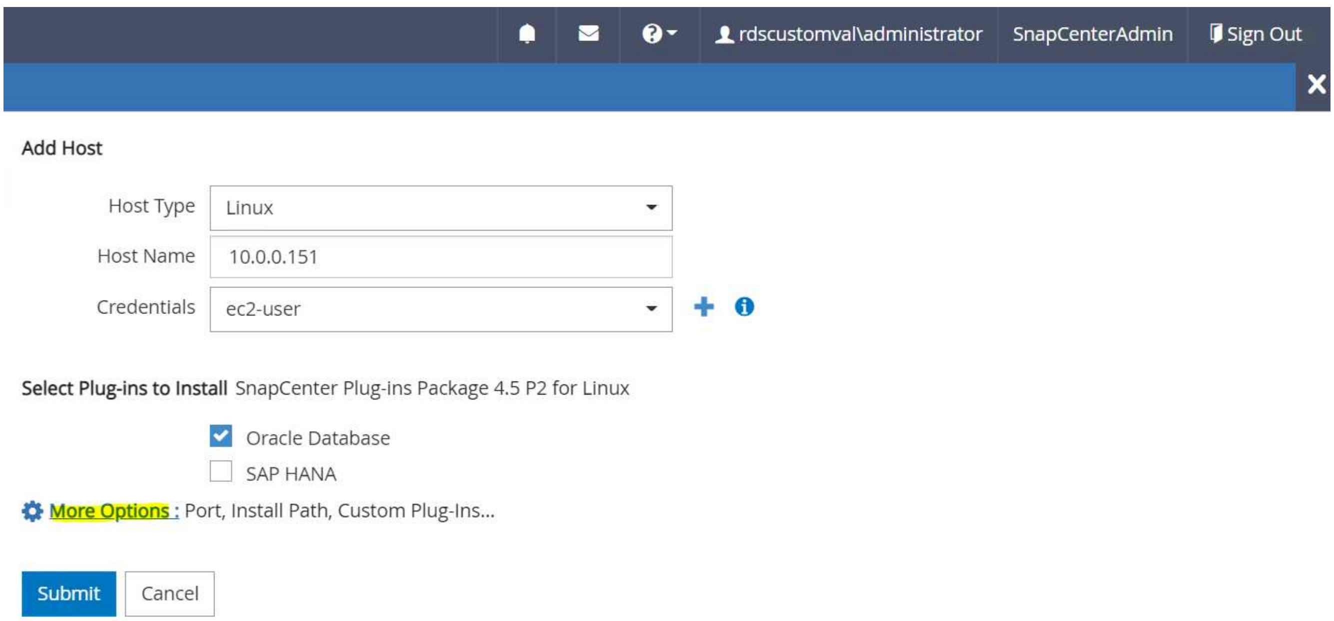
6. On the SnapCenter server Windows host, add the EC2 instance host IP address to the Windows host file `C:\Windows\System32\drivers\etc\hosts`.

```
10.0.0.151      ip-10-0-0-151.ec2.internal
```

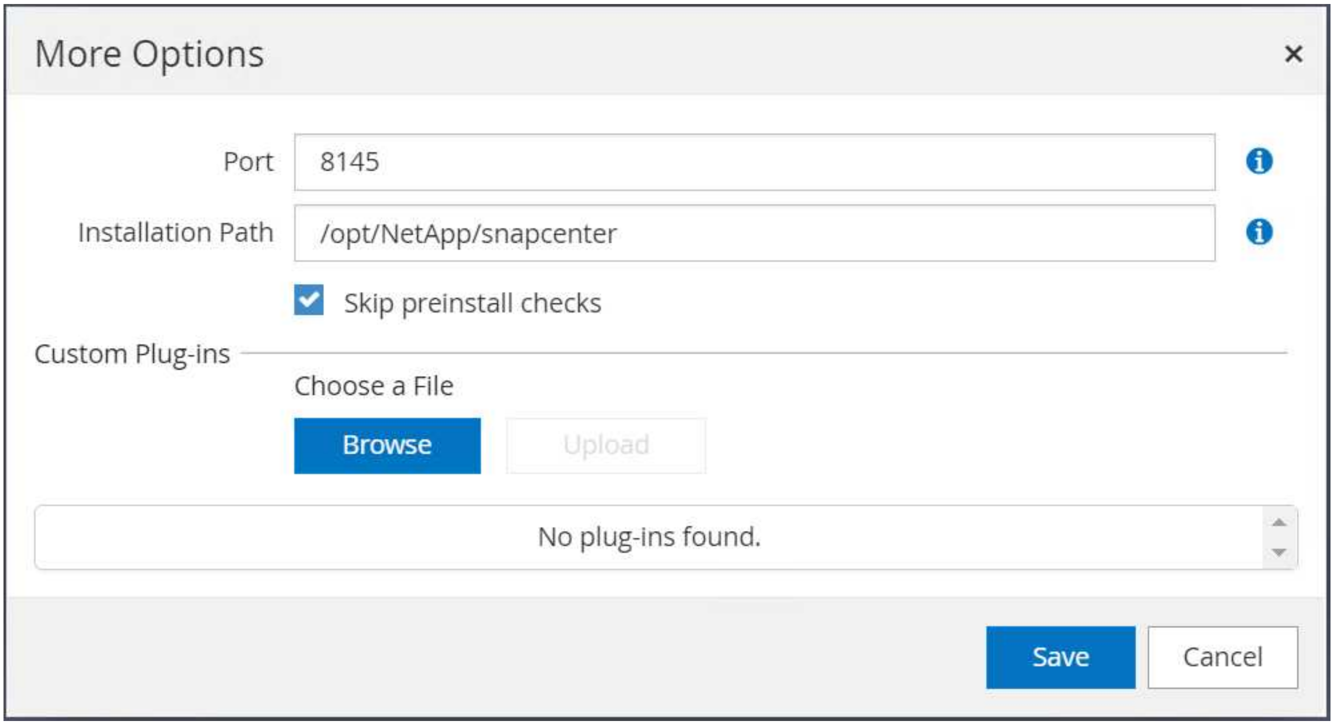
7. In the left-side menu, select Hosts > Managed Hosts, and then click Add to add the EC2 instance host to SnapCenter.



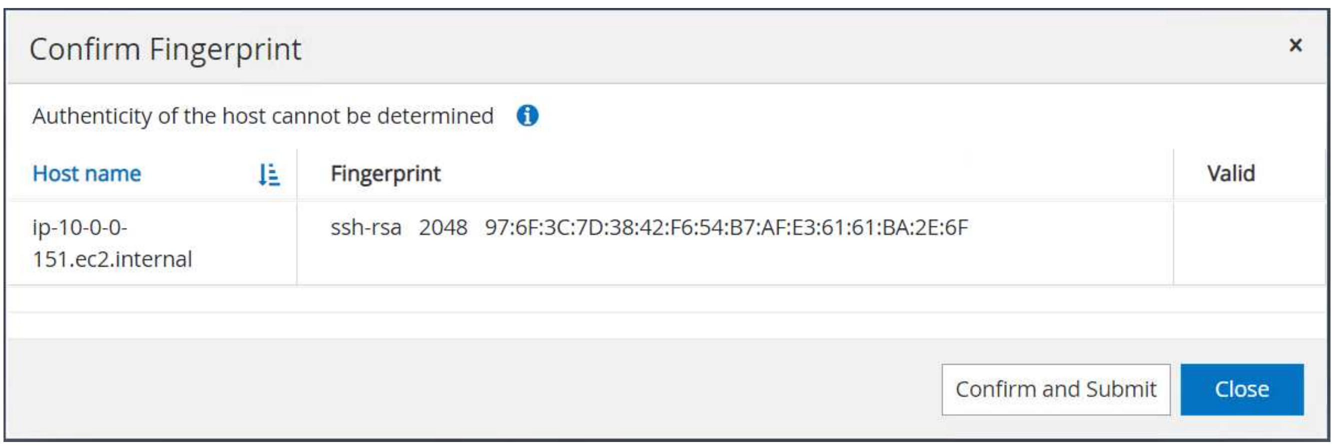

Check Oracle Database, and, before you submit, click More Options.



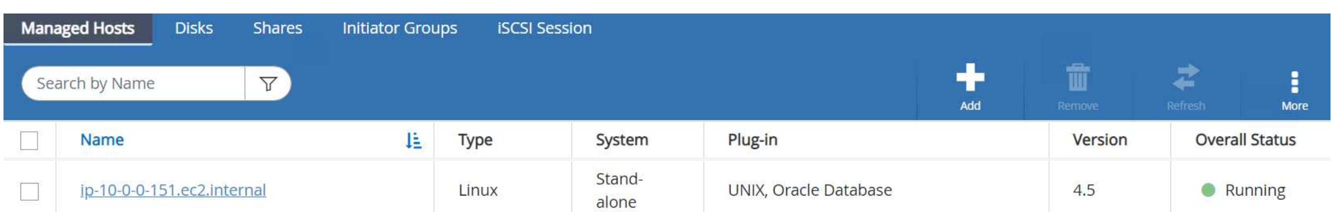

Check Skip Preinstall Checks. Confirm Skipping Preinstall Checks, and then click Submit After Save.




You are prompted with Confirm Fingerprint, and then click Confirm and Submit.




After successful plugin configuration, the managed host's overall status show as Running.




**Configure backup policy for Oracle database**

Refer to this section Setup database backup policy in SnapCenter for details on configuring the Oracle database backup policy.

Generally you need create a policy for the full snapshot Oracle database backup and a policy for the Oracle archive-log-only snapshot backup.

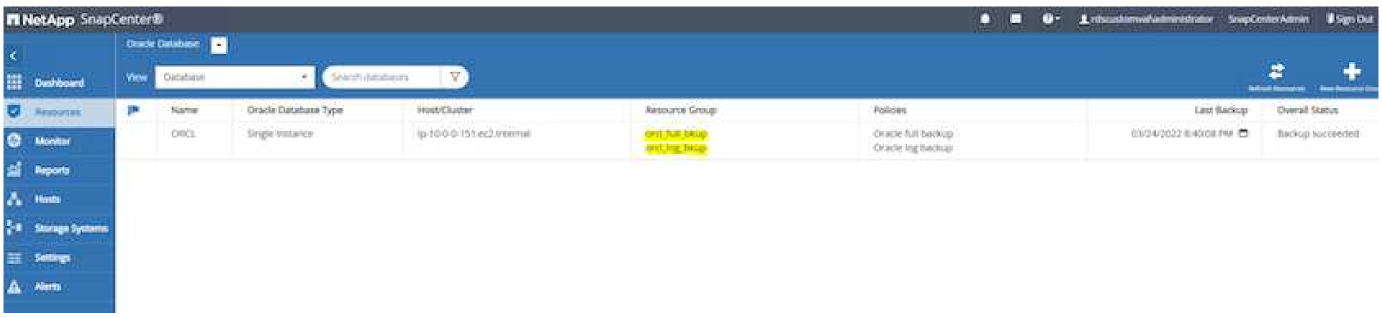
> ⓘ You can enable Oracle archive log pruning in the backup policy to control log-archive space. Check "Update SnapMirror after creating a local Snapshot copy" in "Select secondary replication option" as you need to replicate to a standby location for HA or DR.

**Configure Oracle database backup and scheduling**

Database backup in SnapCenter is user configurable and can be set up either individually or as a group in a resource group. The backup interval depends on the RTO and RPO objectives. NetApp recommends that you run a full database backup every few hours and archive the log backup at a higher frequency such as 10-15 mins for quick recovery.

Refer to the Oracle section of Implement backup policy to protect database for a detailed step-by-step processes for implementing the backup policy created in the section Configure backup policy for Oracle database and for backup job scheduling.

The following image provides an example of the resources groups that are set up to back up an Oracle database.




Next: Database management.

## EC2 and FSx Oracle database management

Previous: Deployment procedures.

### 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.

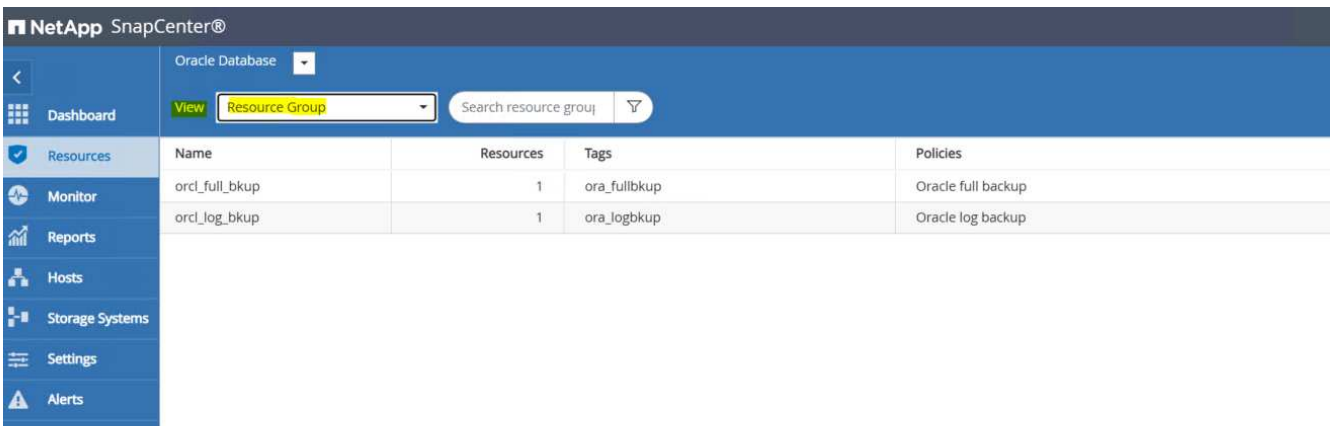
### Taking a snapshot

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.

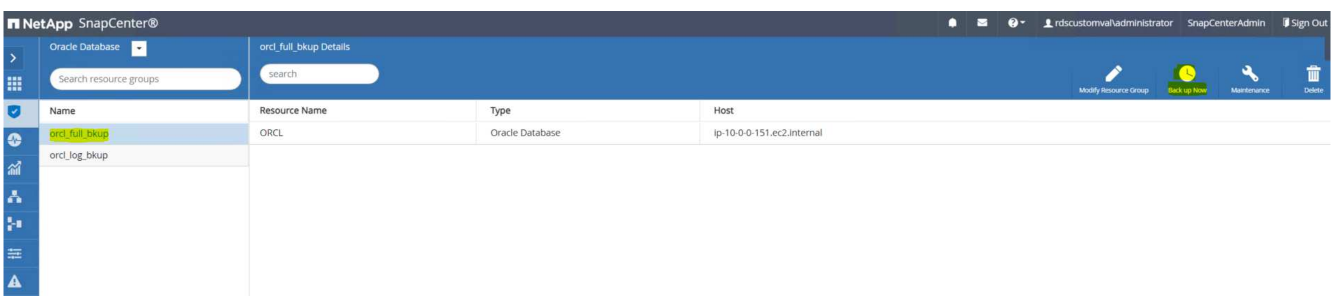
#### Taking a full database snapshot

A full database snapshot includes all Oracle files, including data files, control files, and archive log files.

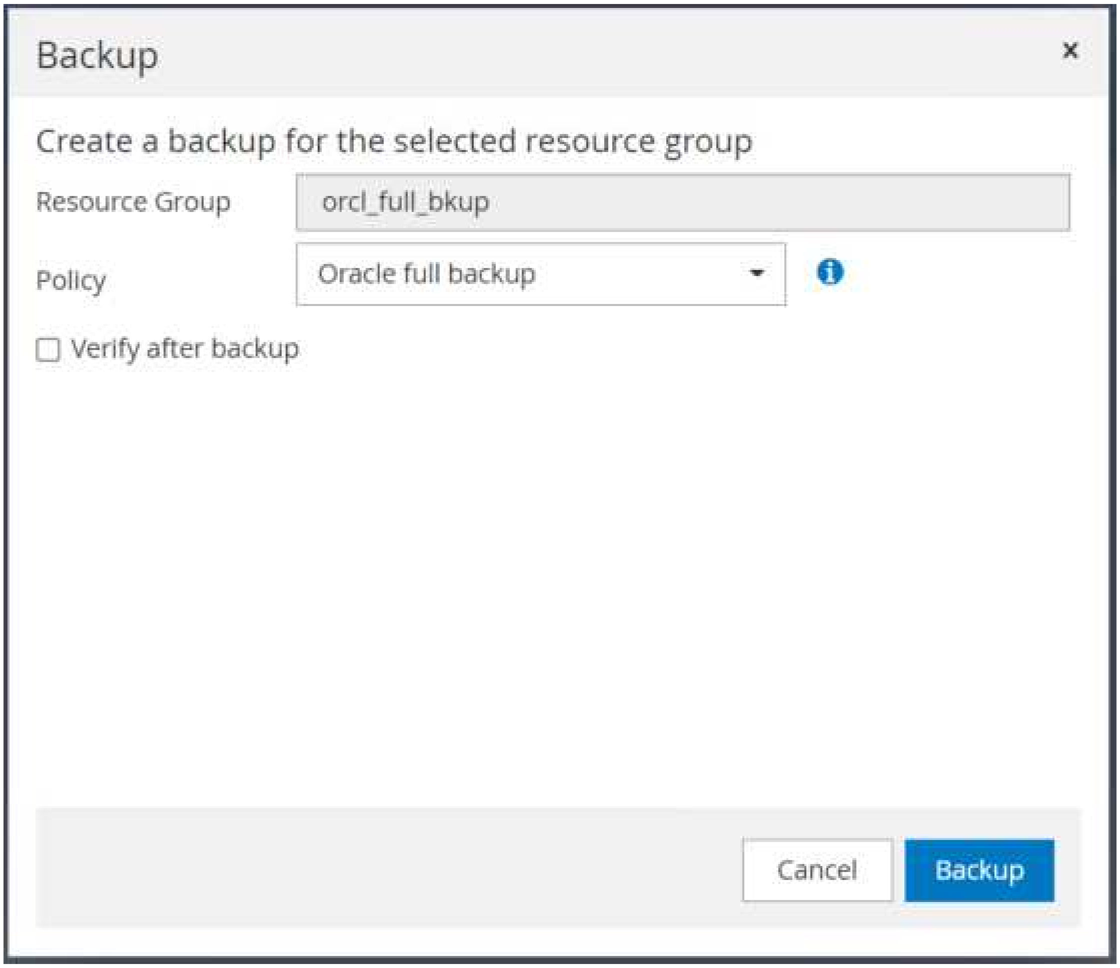
1. Log into the SnapCenter UI and click Resources in the left-side menu. From the View dropdown, change to the Resource Group view.



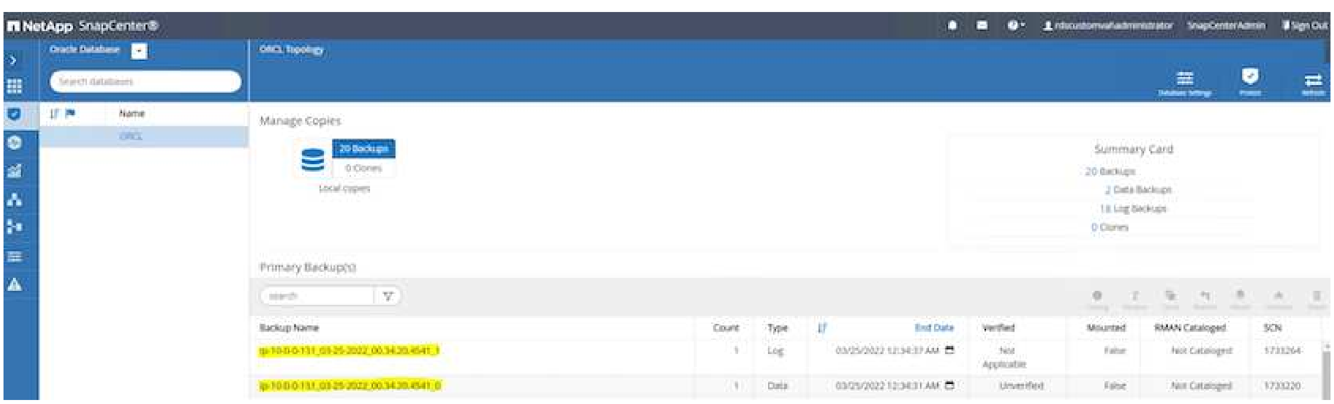

2. Click the full backup resource name, and then click the Backup Now icon to initiate an add-hoc backup.




3. Click Backup and then confirm the backup to start a full database backup.



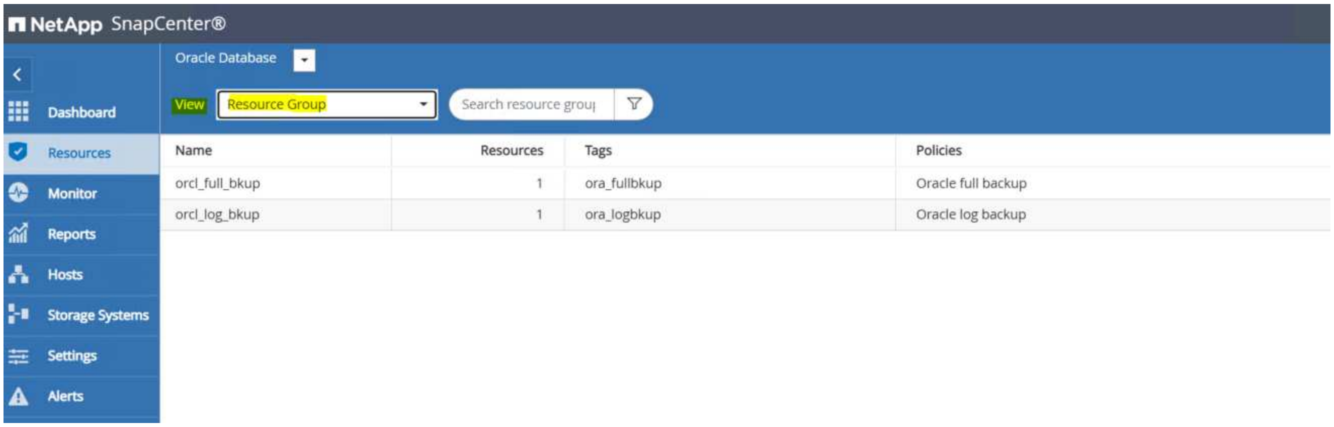

From the Resource view for the database, open the database Managed Backup Copies page to verify that the one-off backup completed successfully. A full database backup creates two snapshots: one for the data volume and one for the log volume.



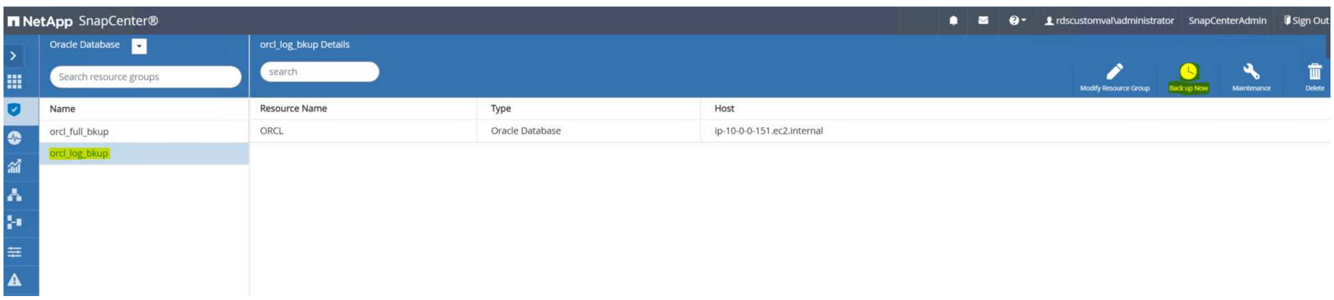

#### Taking an archive log snapshot

An archive log snapshot is only taken for the Oracle archive log volume.

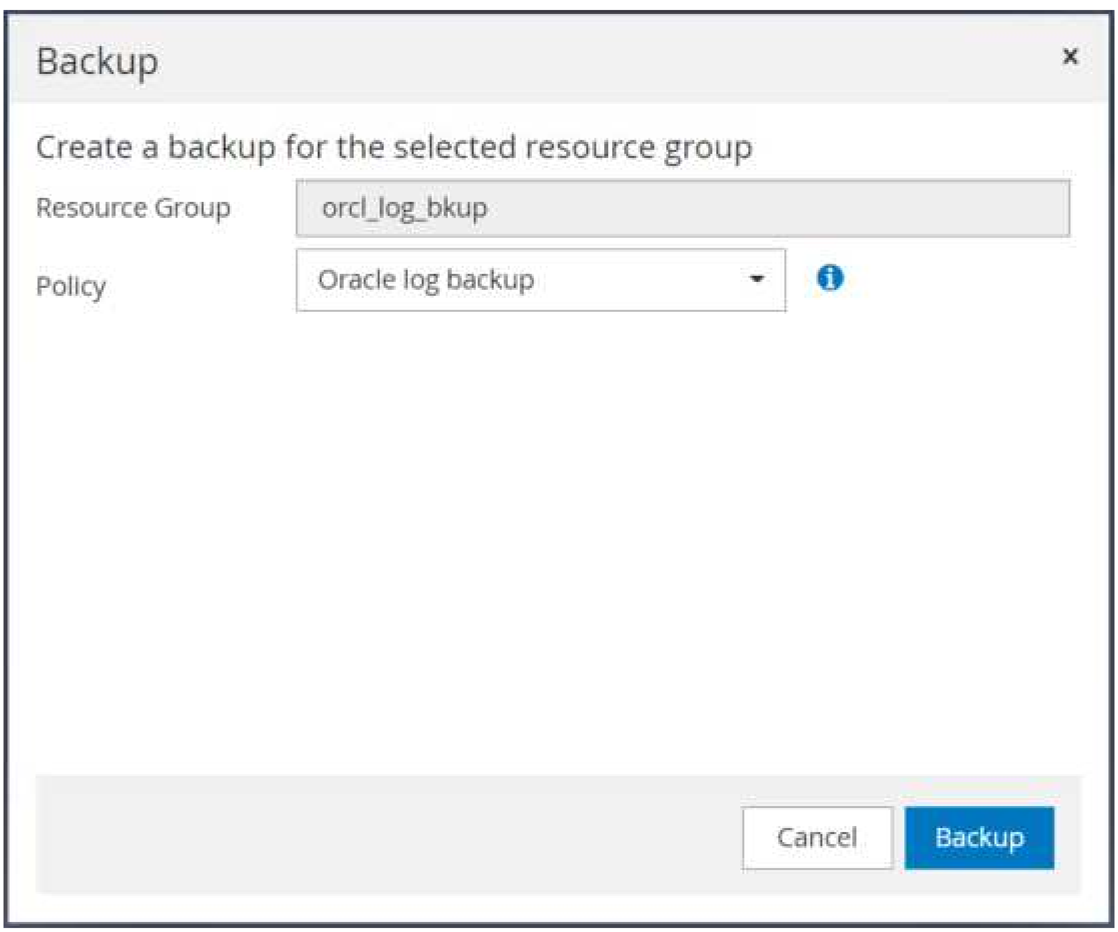
1. Log into the SnapCenter UI and click the Resources tab in the left-side menu bar. From the View dropdown, change to the Resource Group view.



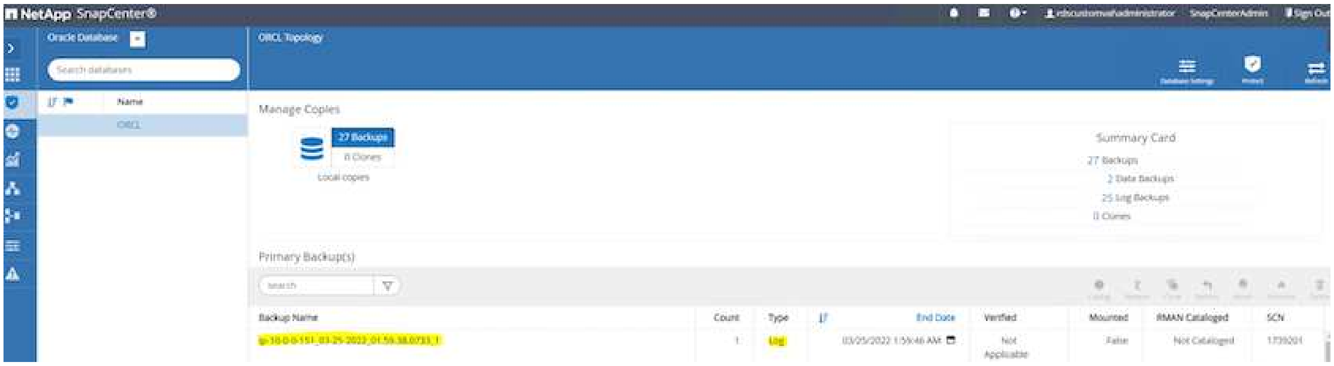

2. Click the log backup resource name, and then click the Backup Now icon to initiate an add-hoc backup for archive logs.




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.

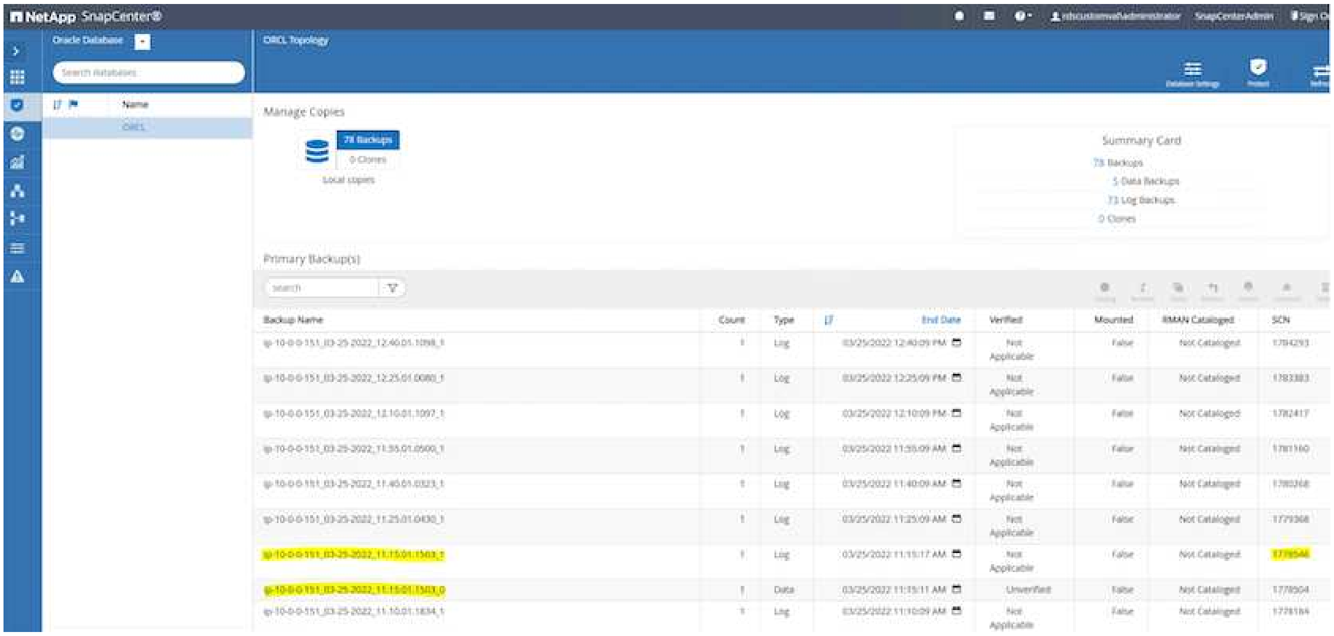


### Restoring to a point in time

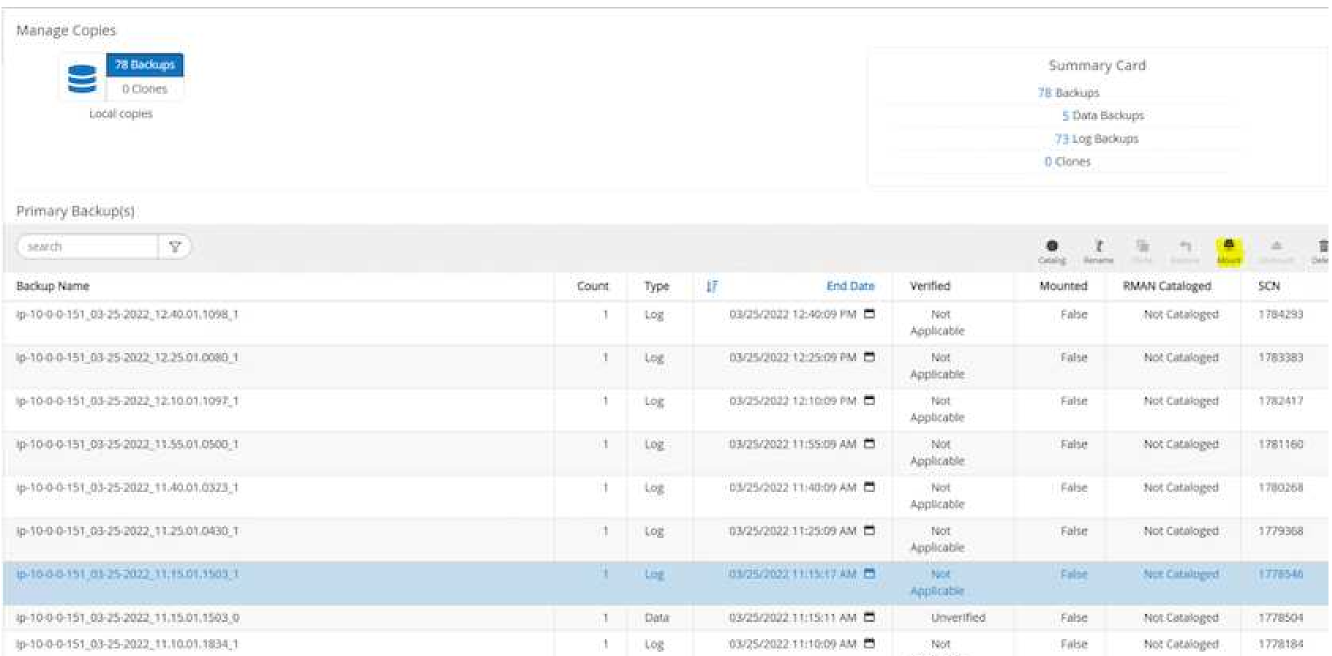
SnapCenter-based restore to a point in time is executed on the same EC2 instance host. Complete the following steps to perform the restore:

1. From the SnapCenter Resources tab > Database view, click the database name to open the database backup.

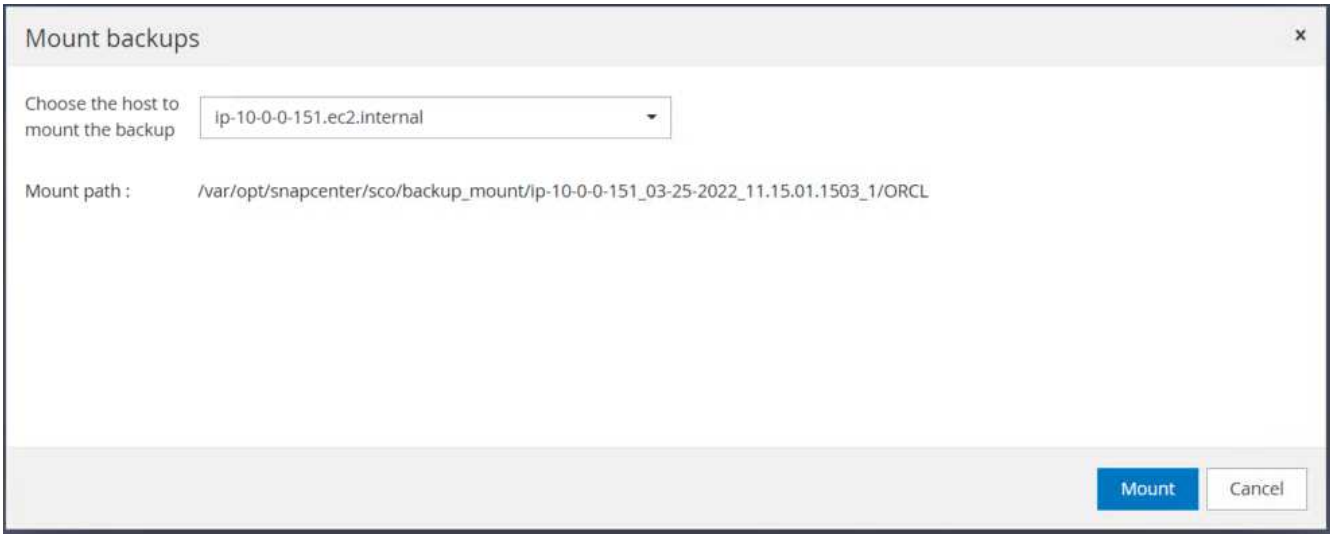
2. Select the database backup copy and the desired point in time to be restored. Also mark down the corresponding SCN number for the point in time. The point-in-time restore can be performed using either the time or the SCN.



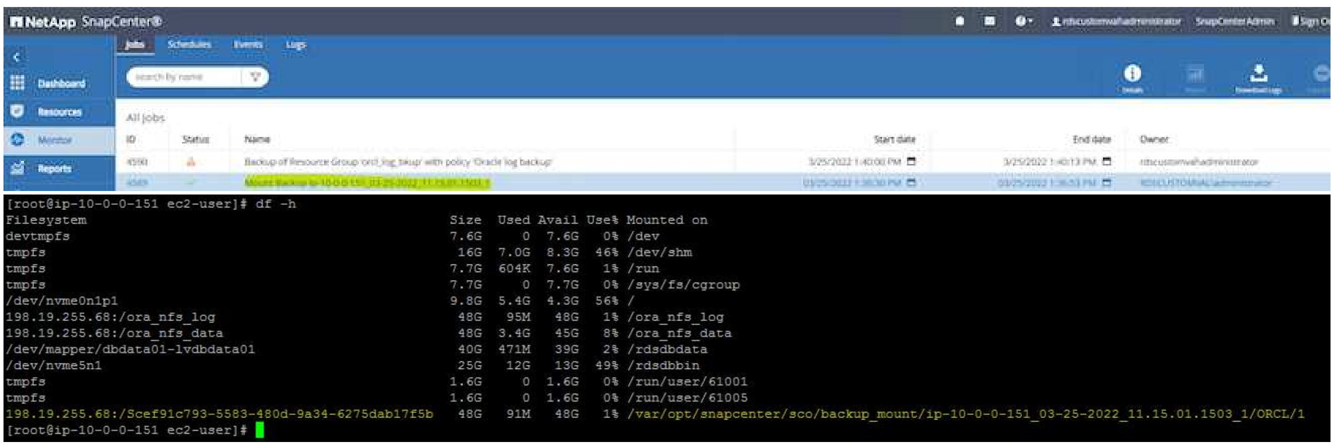

3. Highlight the log volume snapshot and click the Mount button to mount the volume.




4. Choose the primary EC2 instance to mount the log volume.



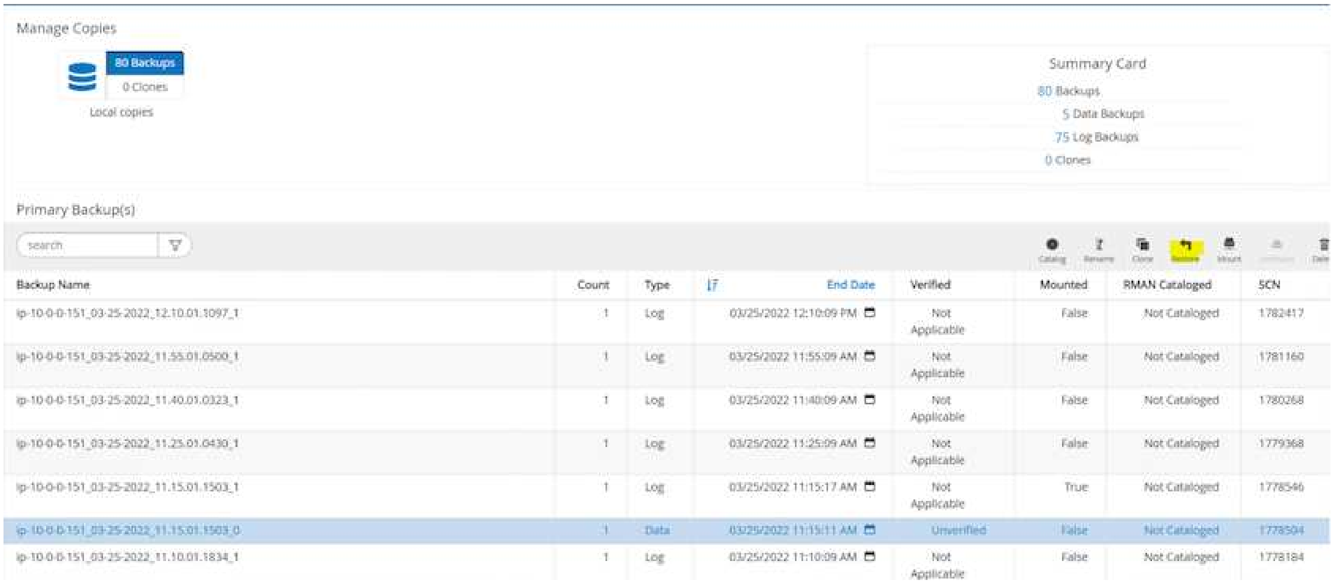

5. Verify that the mount job completes successfully. Also check on the EC2 instance host to see the that log volume mounted and also the mount point path.



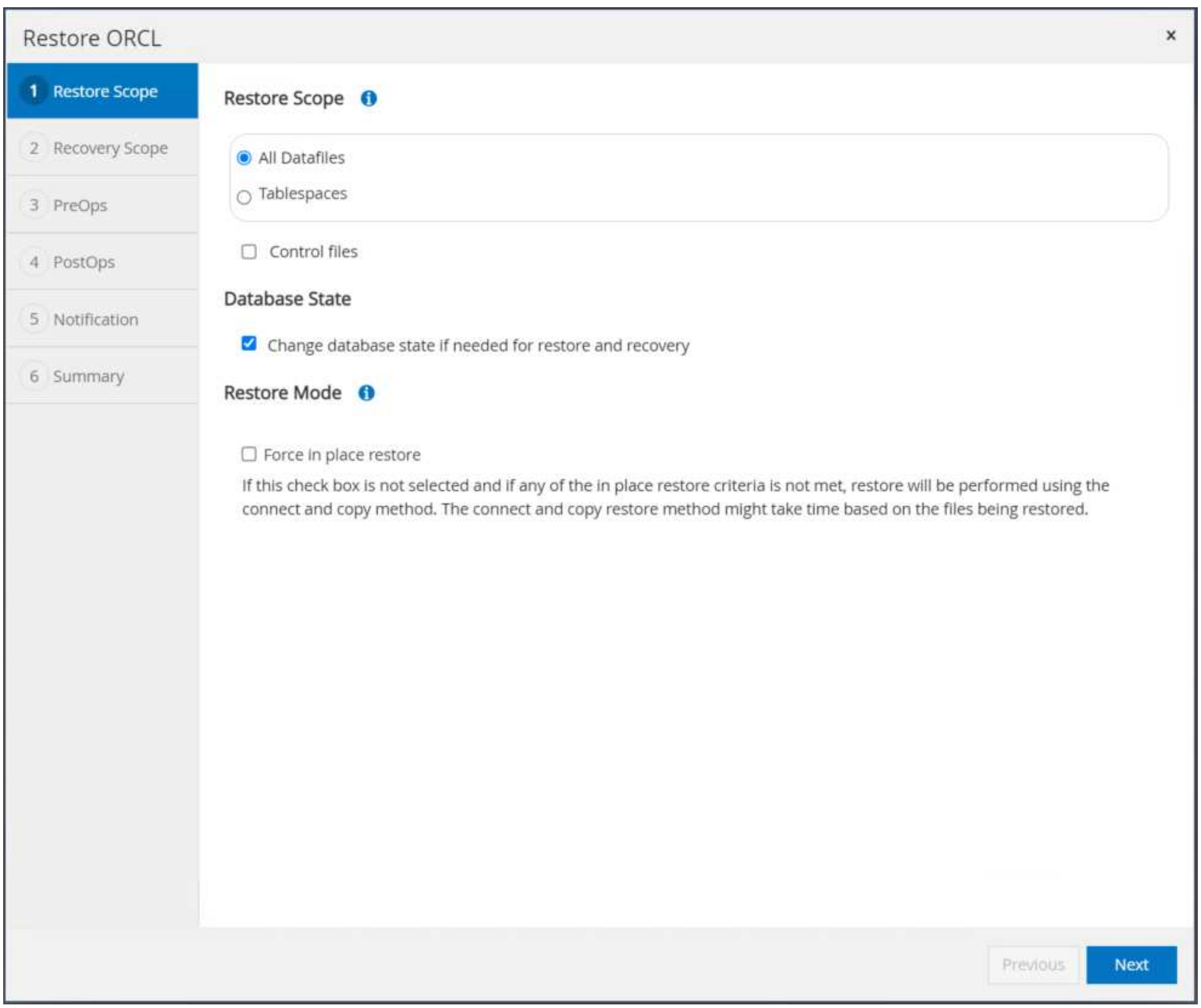

6. Copy the archive logs from the mounted log volume to the current archive log directory.

```
[ec2-user@ip-10-0-0-151 ~]$ cp /var/opt/snapcenter/sco/backup_mount/ip-
10-0-0-151_03-25-2022_11.15.01.1503_1/ORCL/1/db/ORCL_A/arch/*.arc
/ora_nfs_log/db/ORCL_A/arch/
```

7. Return to the SnapCenter Resource tab > database backup page, highlight the data snapshot copy, and click the Restore button to start the database restore workflow.

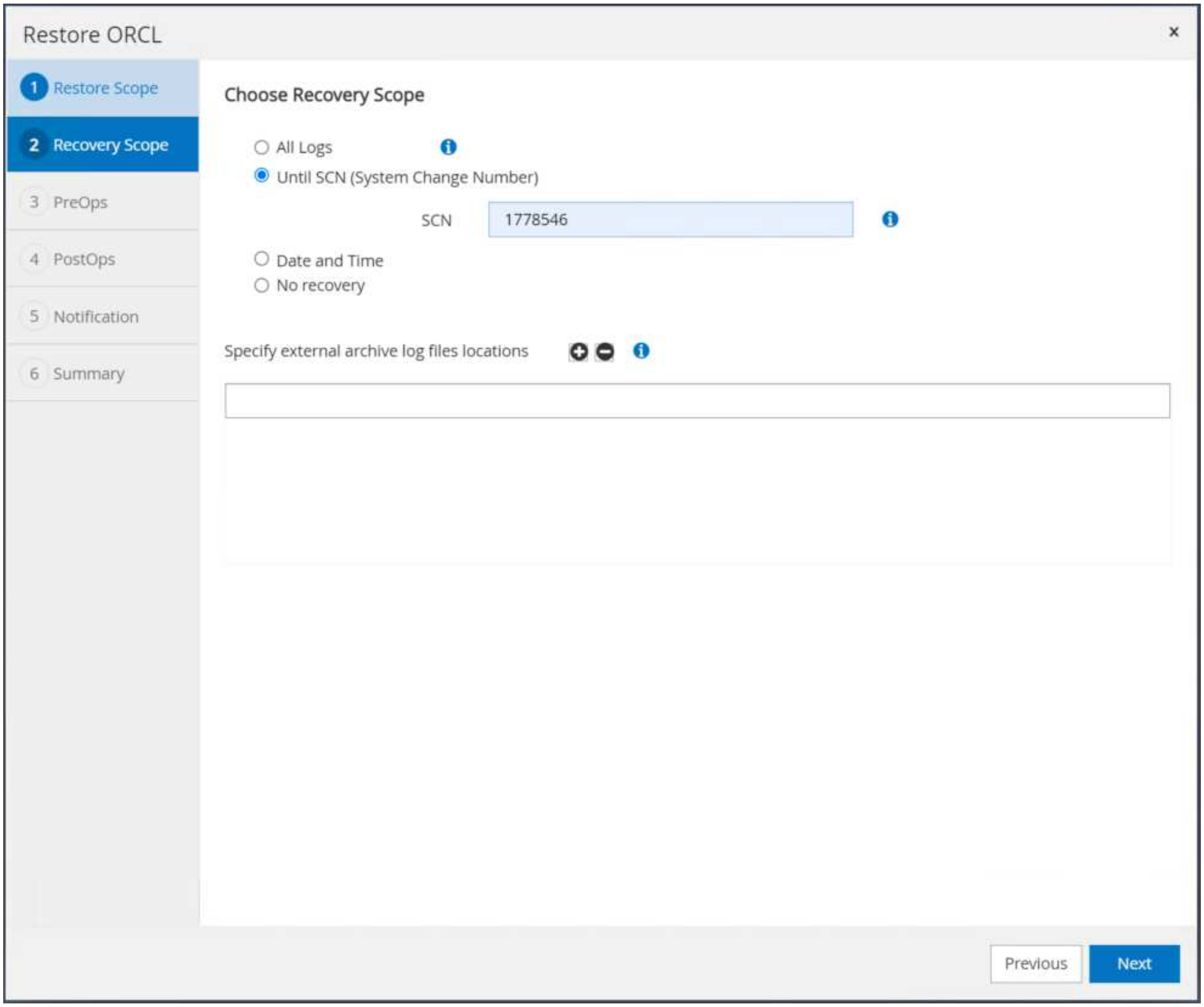



8. Check "All Datafiles" and "Change database state if needed for restore and recovery", and click Next.

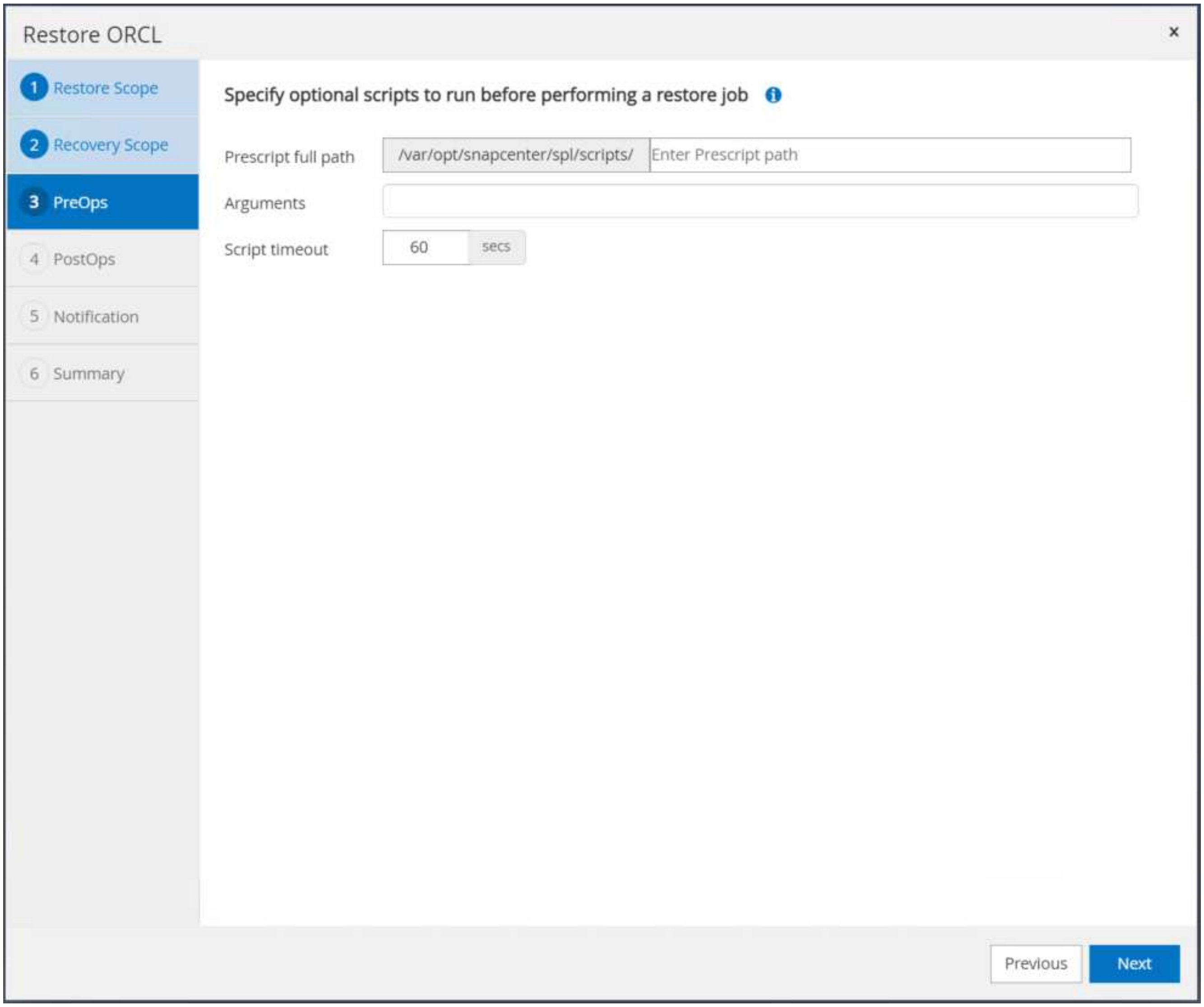



9. Choose a desired recovery scope using either SCN or time. Rather than copying the mounted archive logs

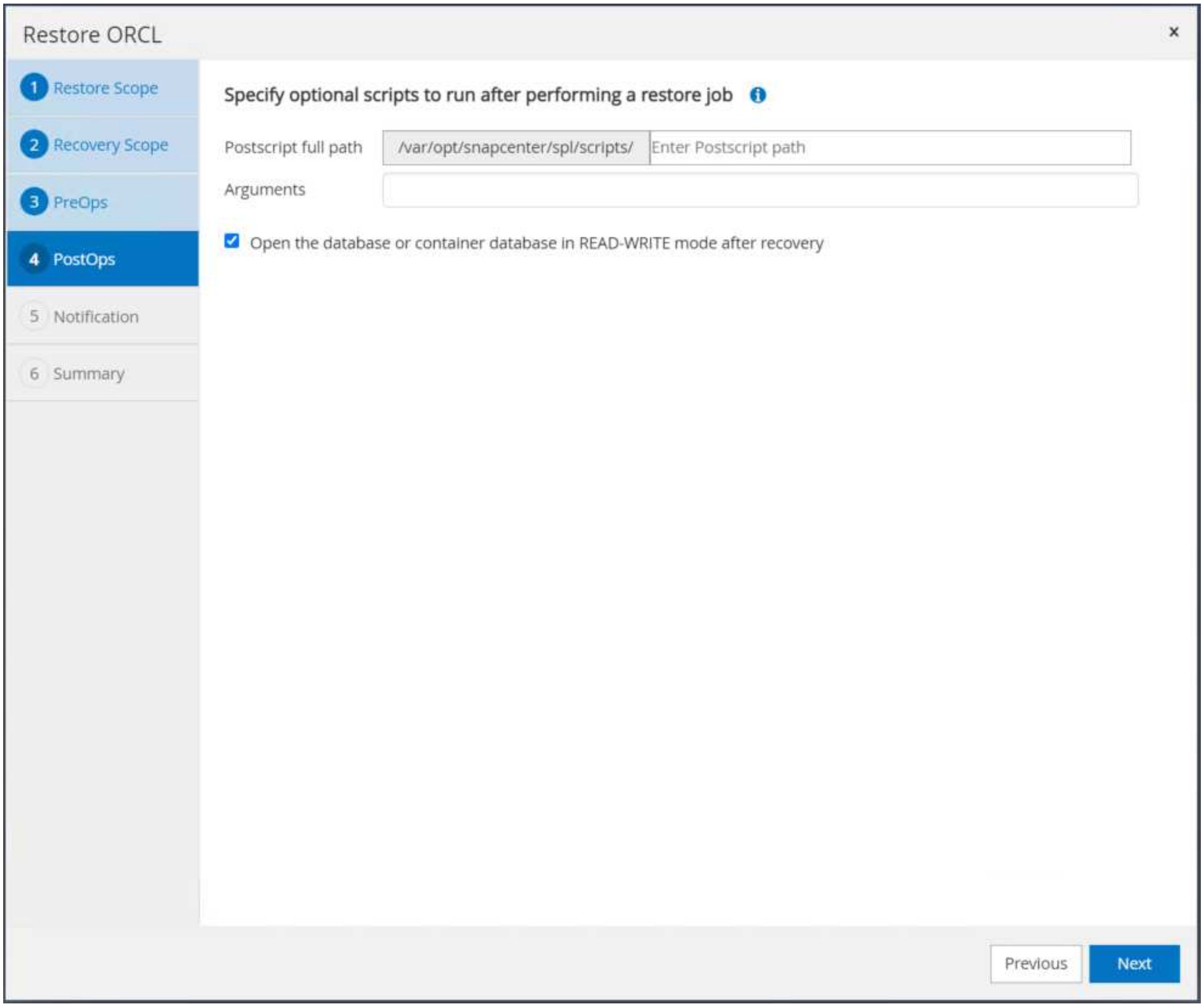
to the current log directory as demonstrated in step 6, the mounted archive log path can be listed in "Specify external archive log files locations" for recovery.



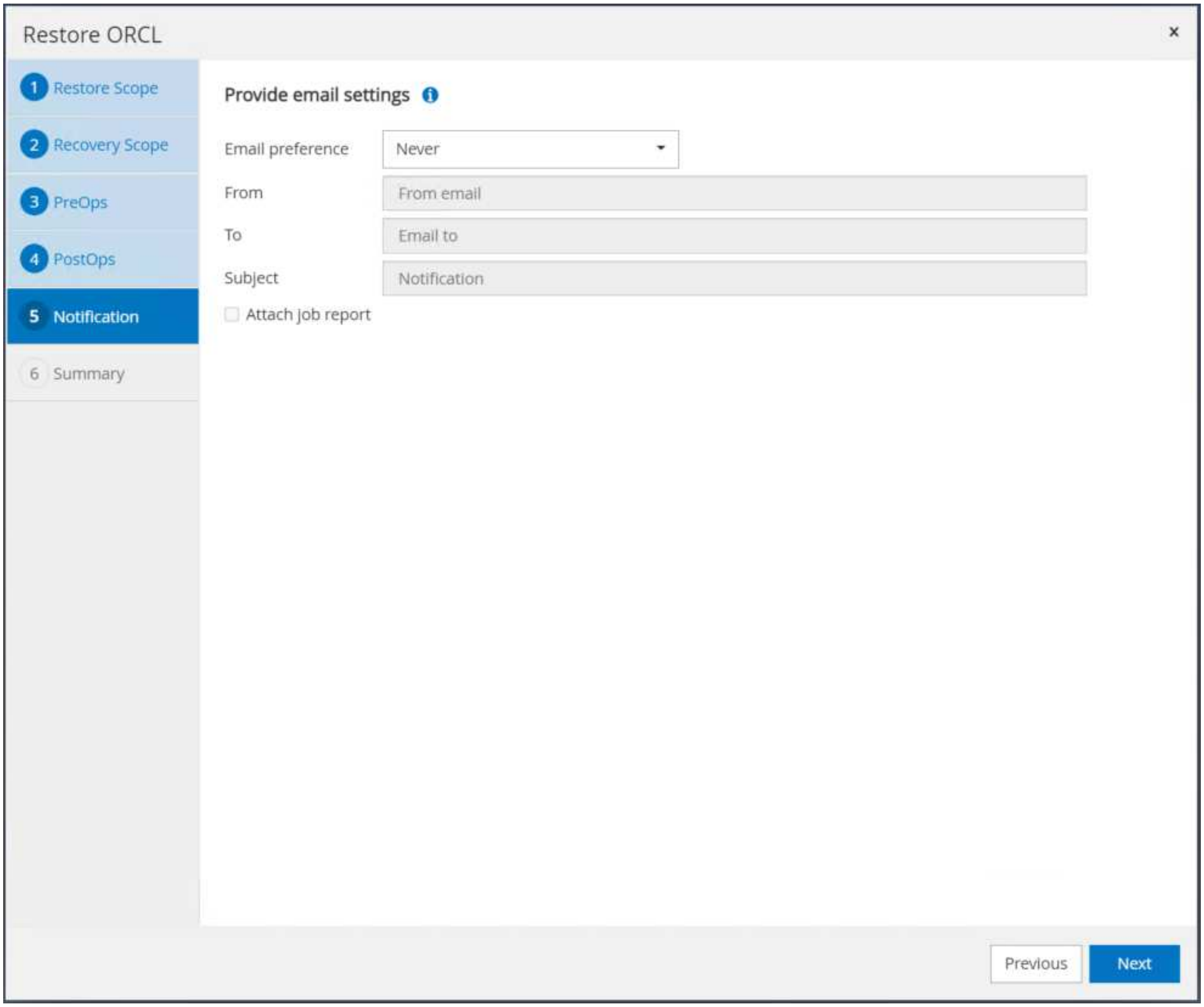

10. Specify an optional prescript to run if necessary.



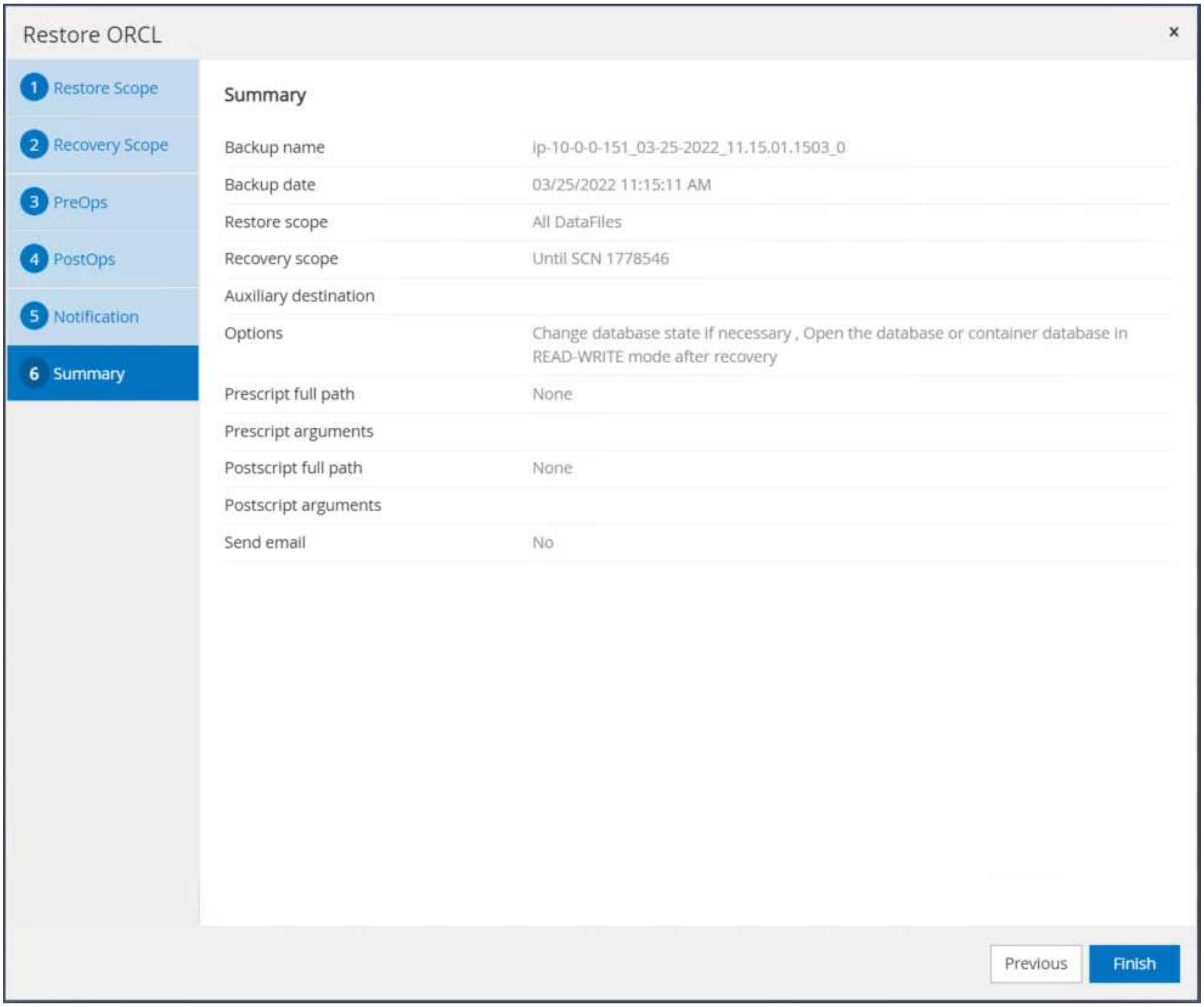

11. Specify an optional afterscript to run if necessary. Check the open database after recovery.



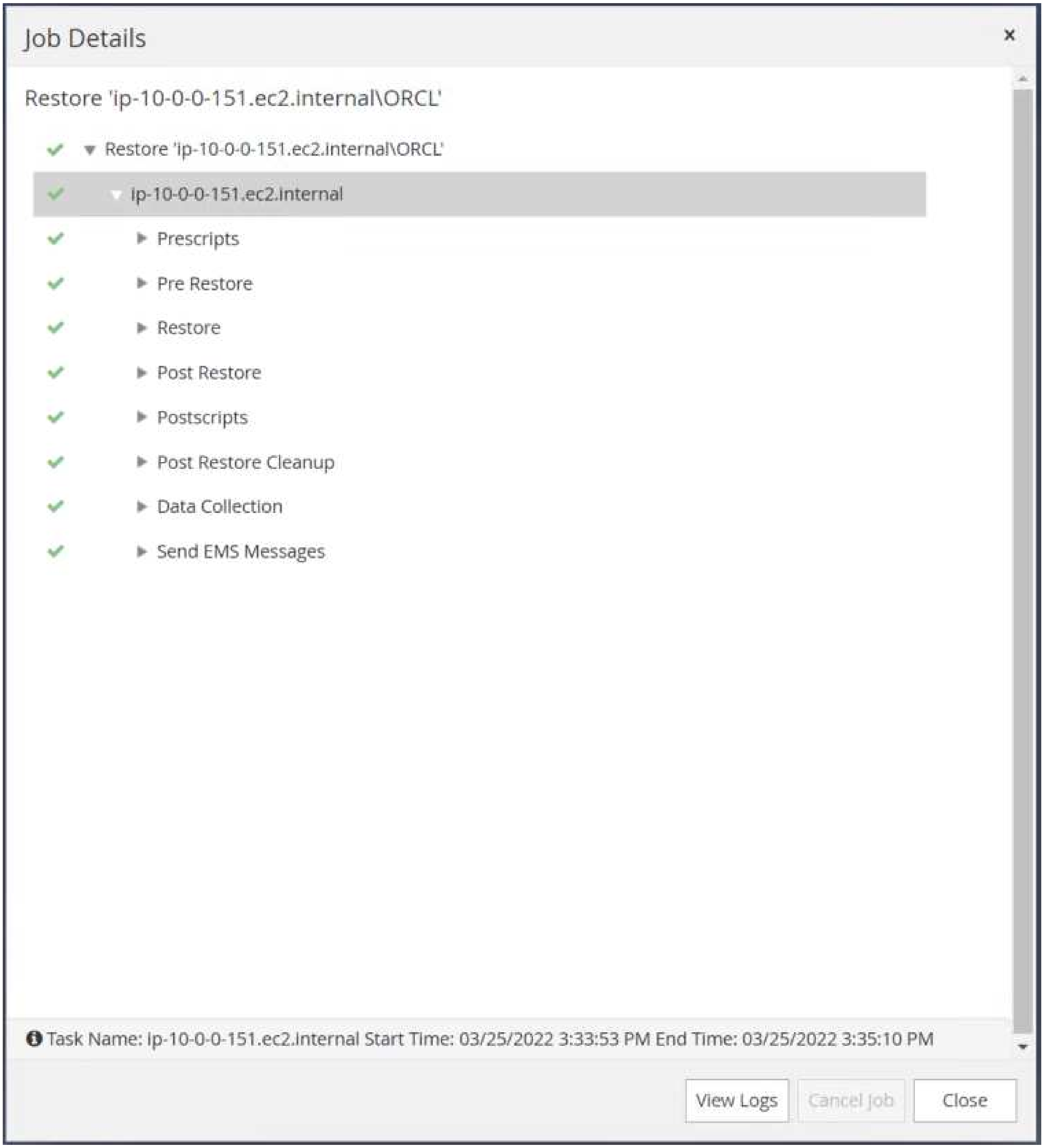

12. Provide an SMTP server and email address if a job notification is needed.




13. Restore the job summary. Click finish to launch the restore job.




14. Validate the restore from SnapCenter.




15. Validate the restore from the EC2 instance host.

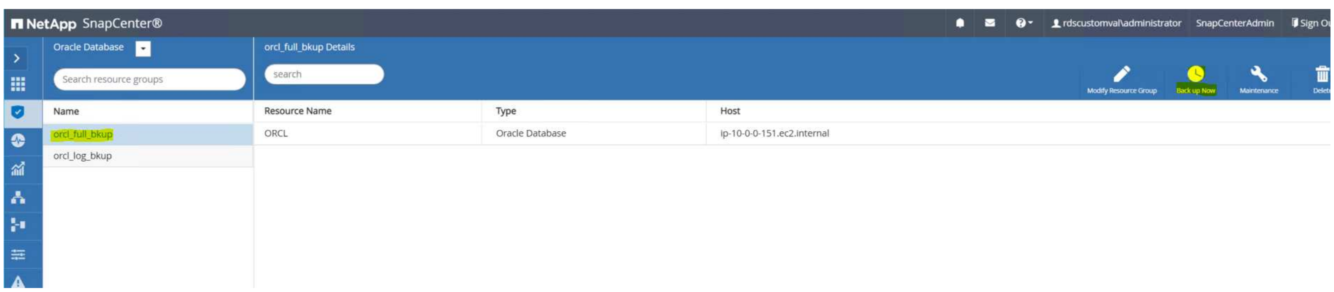
```
-bash-4.2$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Mar 25 15:44:08 2022
Version 19.8.0.0.0

Copyright (c) 1982, 2020, 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> select name, RESETLOGS_CHANGE#, RESETLOGS_TIME, open_mode from v$database;

NAME      RESETLOGS_CHANGE# RESETLOGS OPEN_MODE
--------- ----------------- --------- --------------------
ORCL                1778547 25-MAR-22 READ WRITE

SQL>
```

16. To unmount the restore log volume, reverse the steps in step 4.

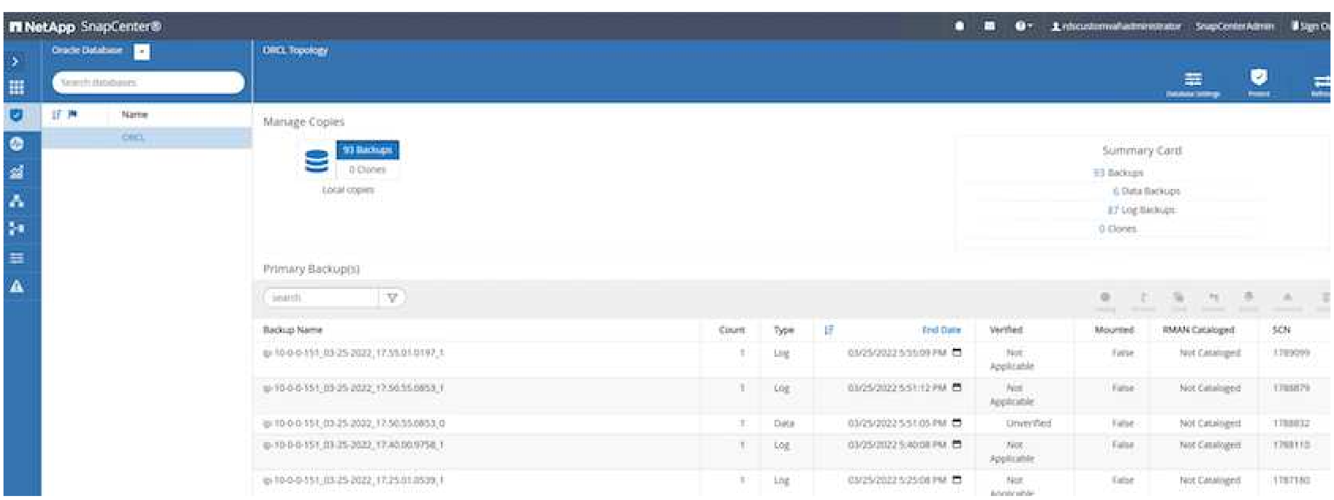
### Creating a database clone

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.

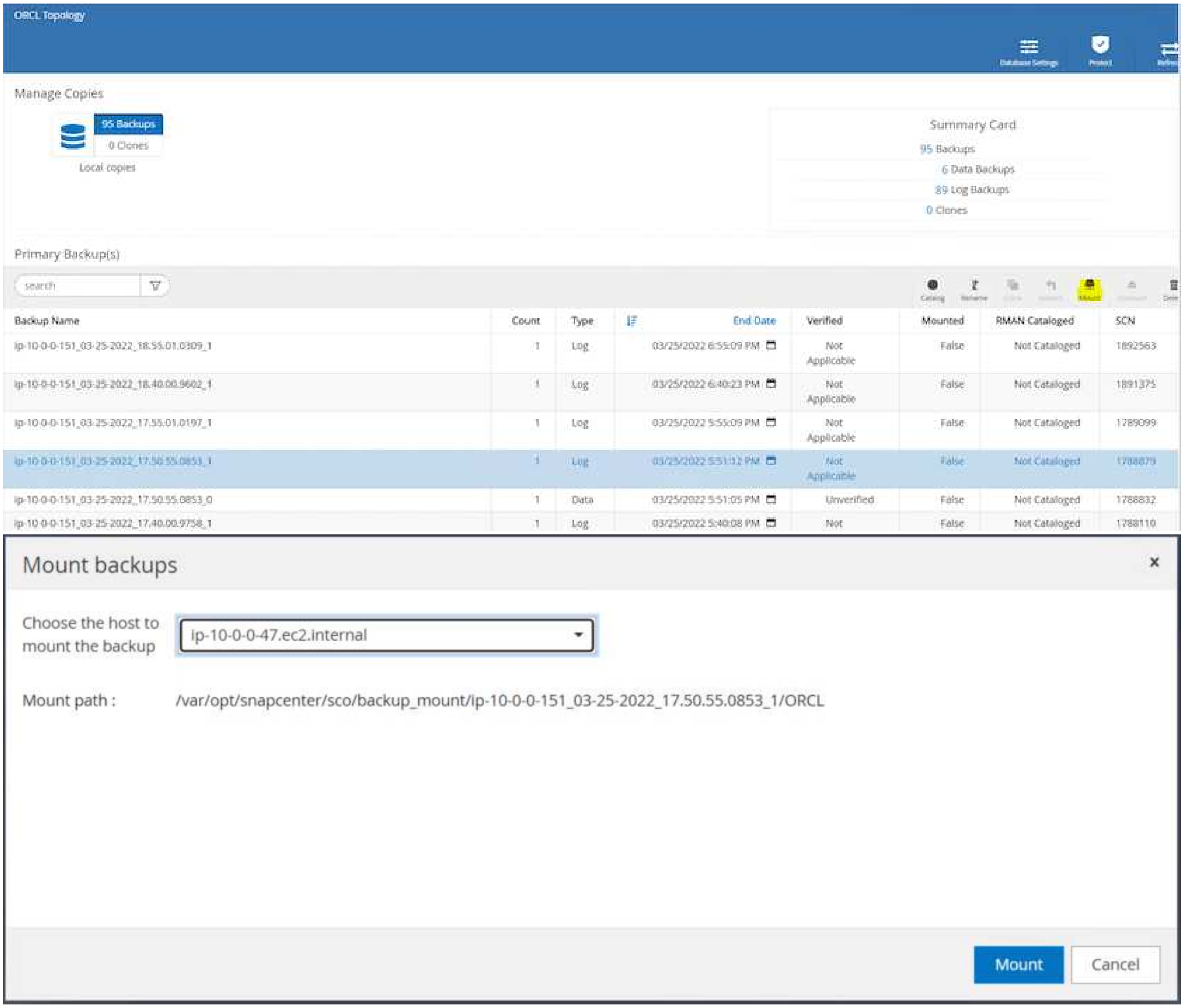
1. Take a full snapshot backup of the primary database from SnapCenter using the full backup resource group.



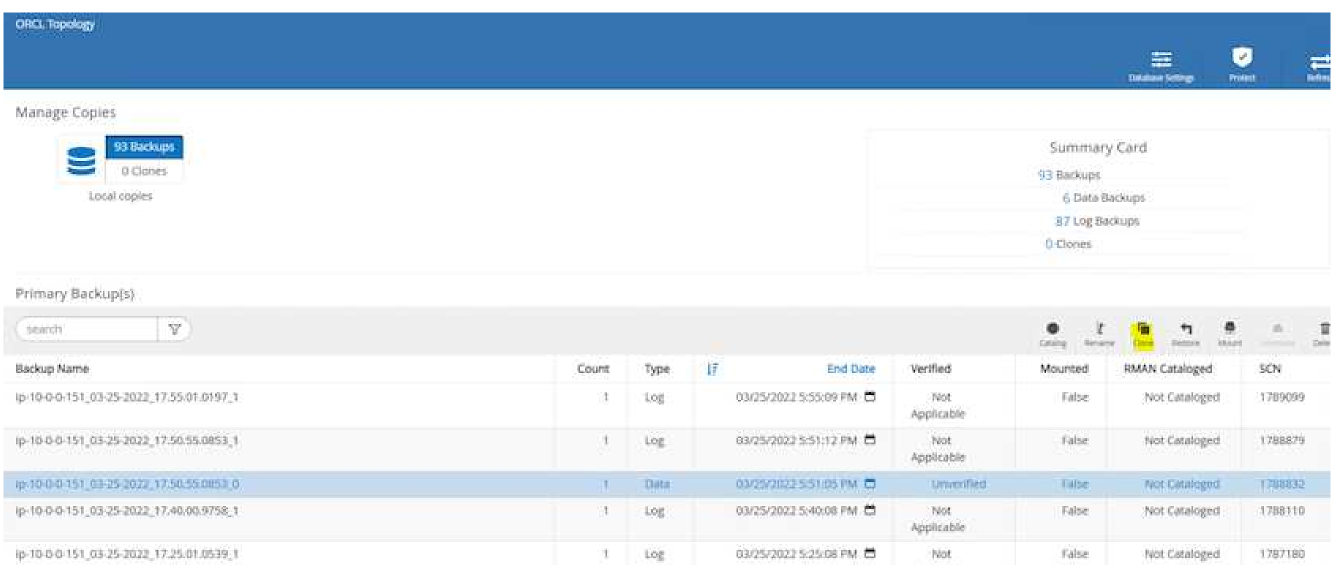

2. From the SnapCenter Resource tab > Database view, open the Database Backup Management page for the primary database that the replica is to be created from.



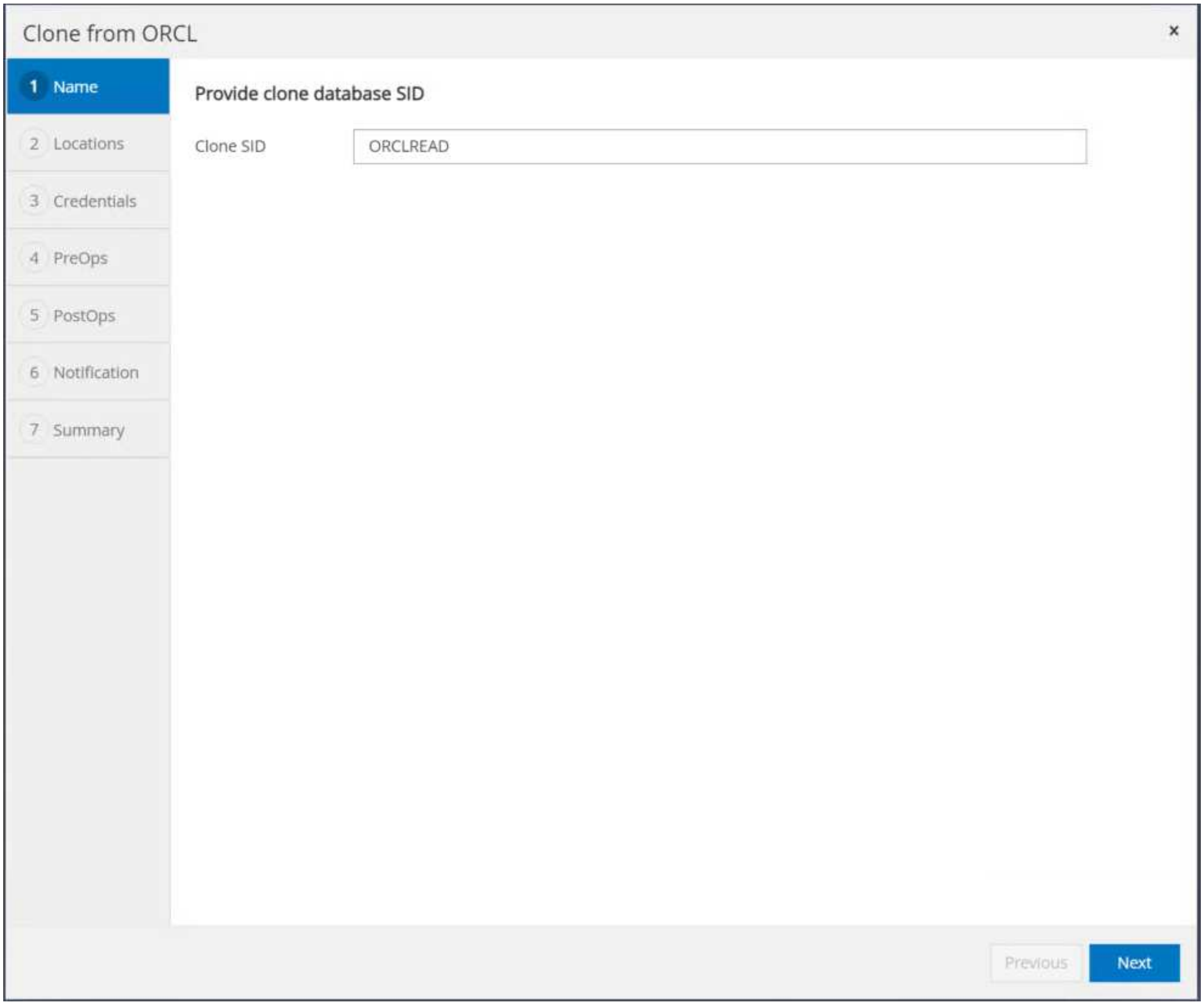

3. Mount the log volume snapshot taken in step 4 to the standby EC2 instance host.



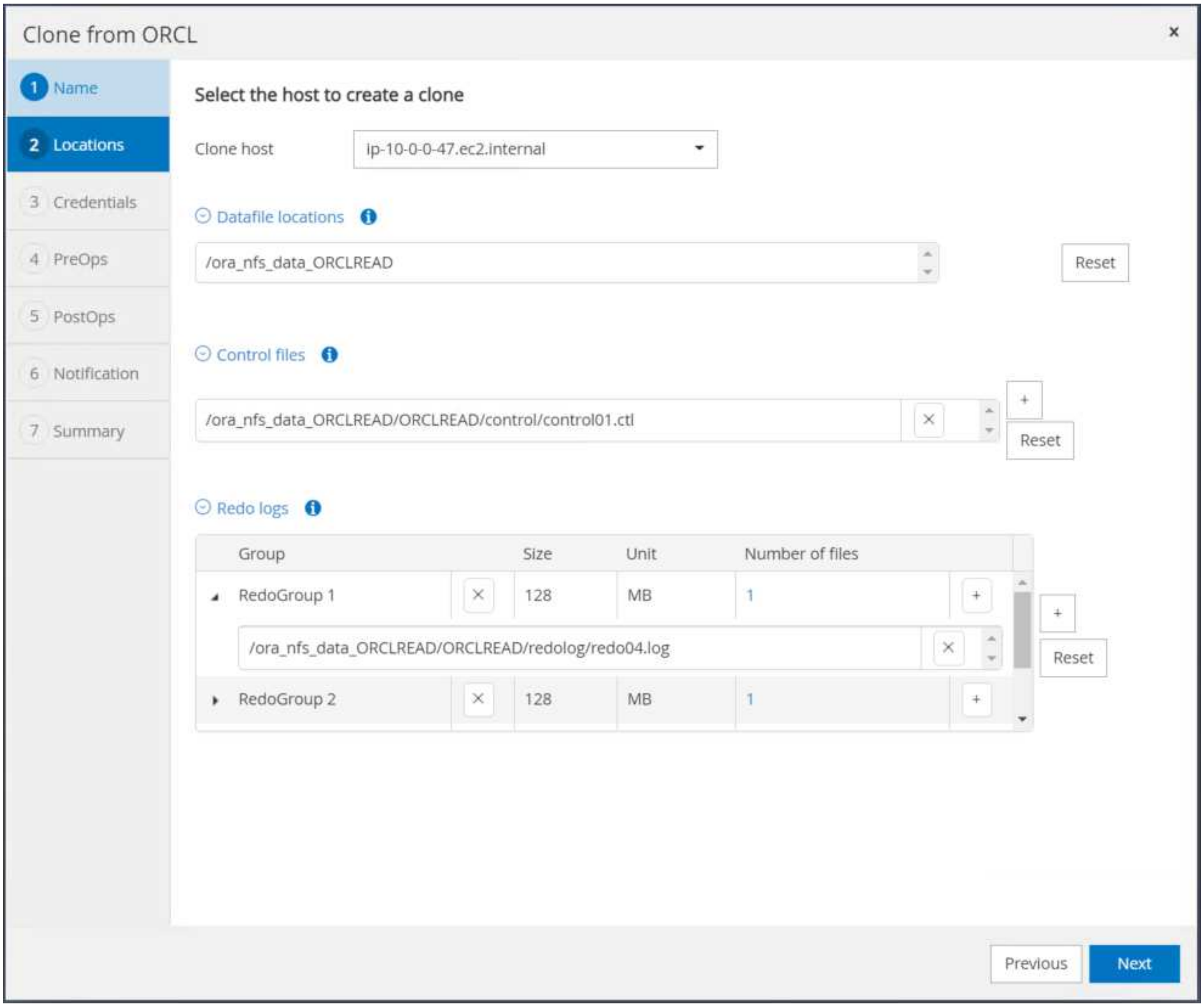
4. Highlight the snapshot copy to be cloned for the replica, and click the Clone button to start the clone procedure.



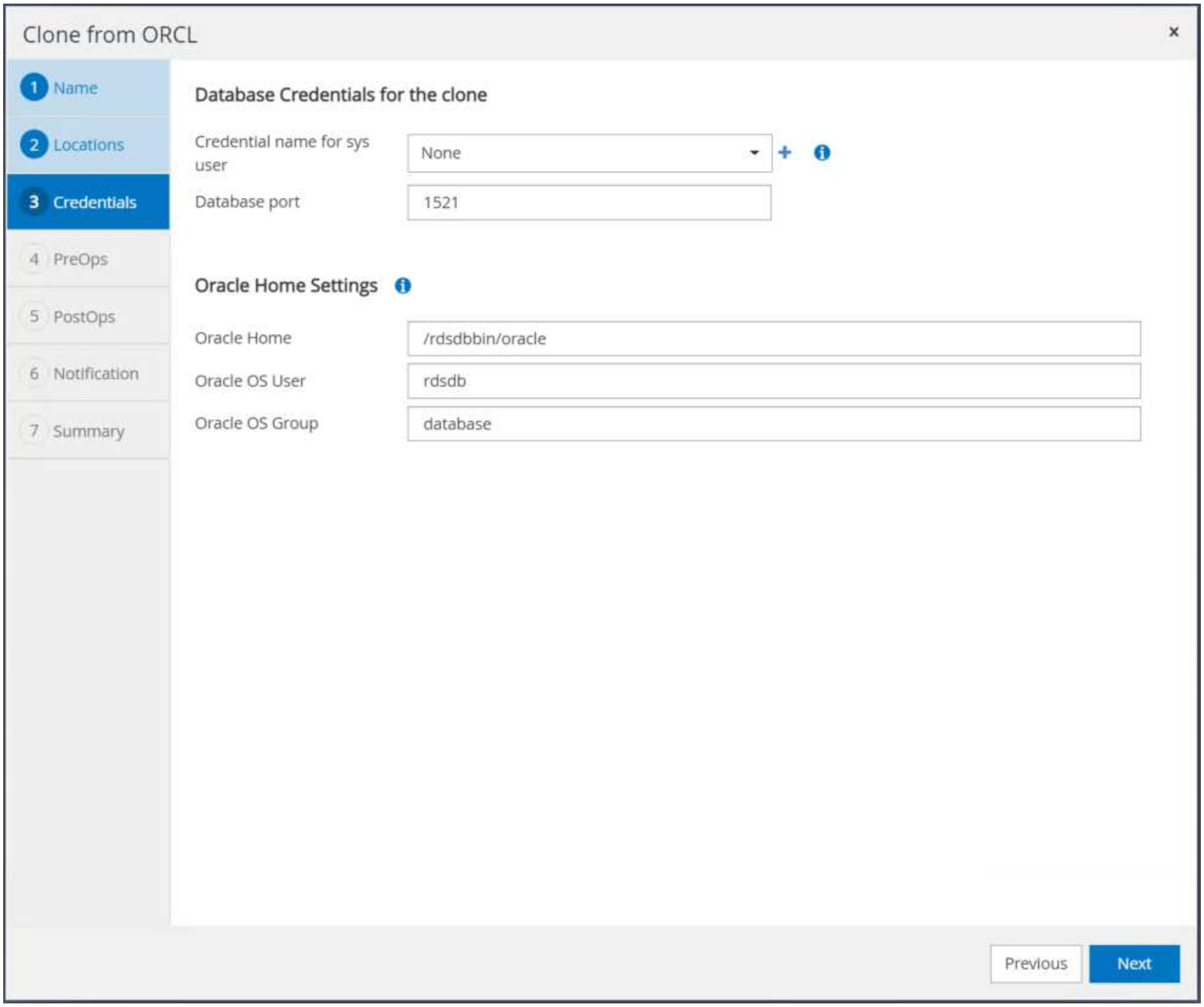
5. Change the replica copy name so that it is different from the primary database name. Click Next.



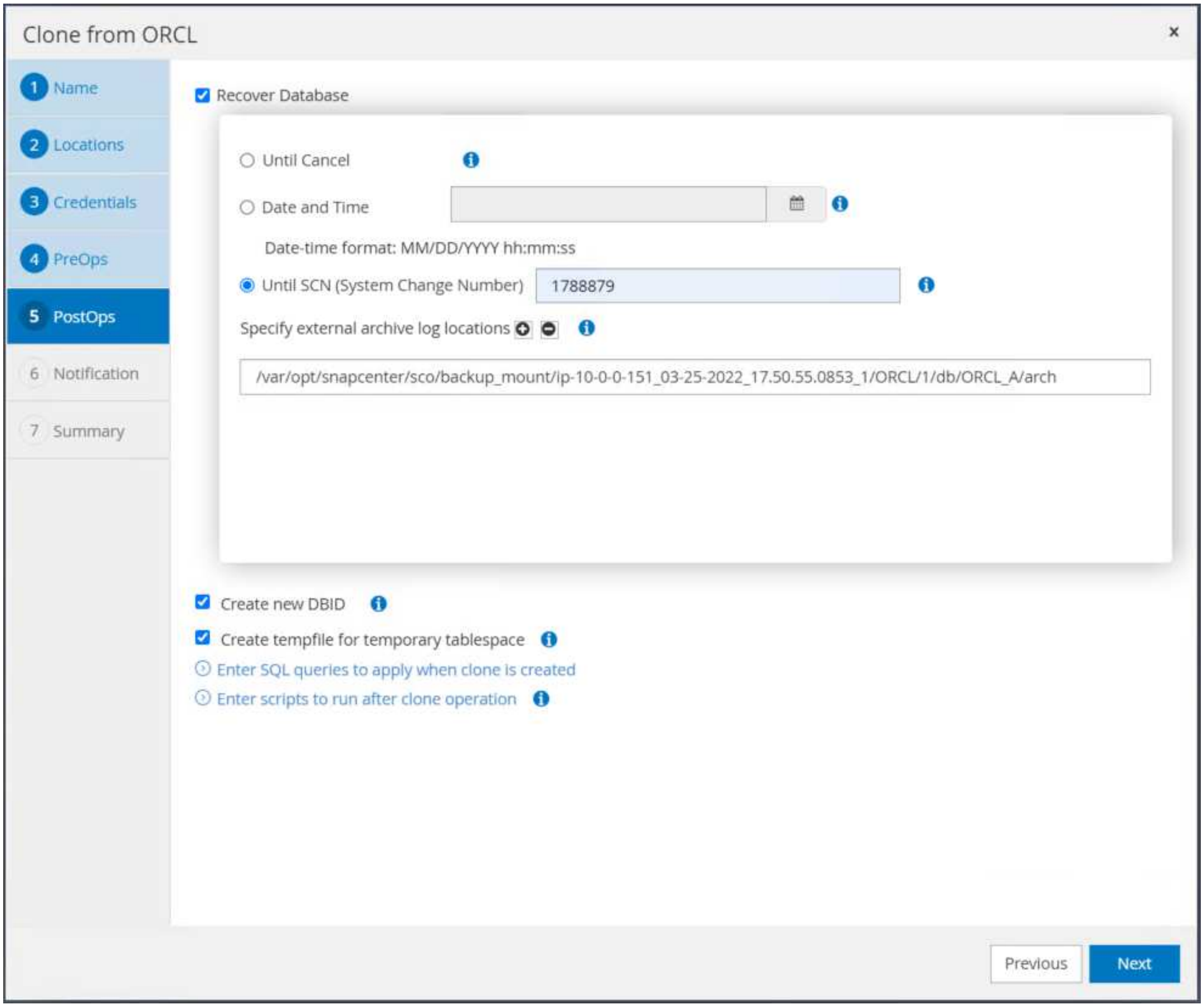

6. Change the clone host to the standby EC2 host, accept the default naming, and click Next.



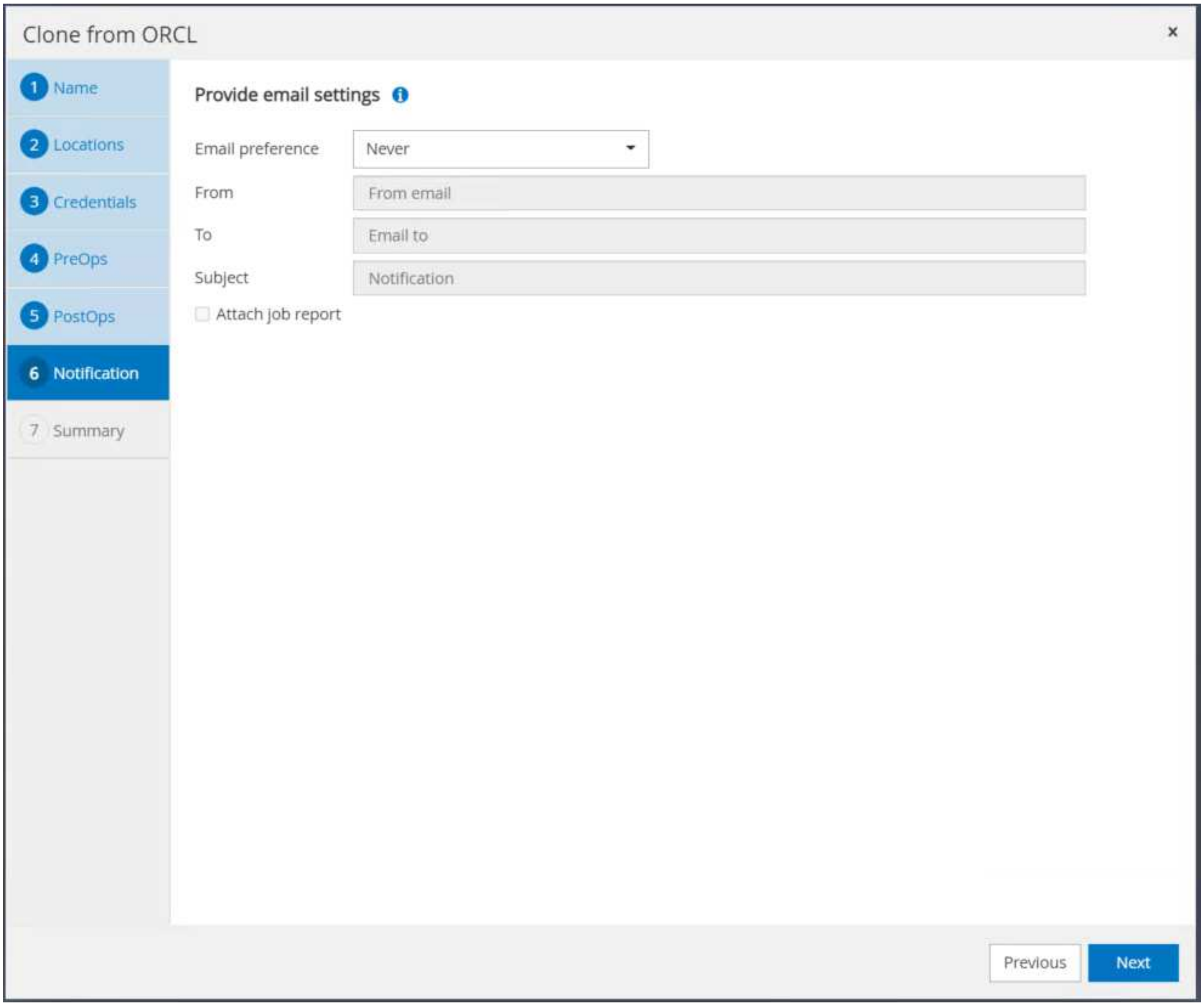

7. Change your Oracle home settings to match those configured for the target Oracle server host, and click Next.



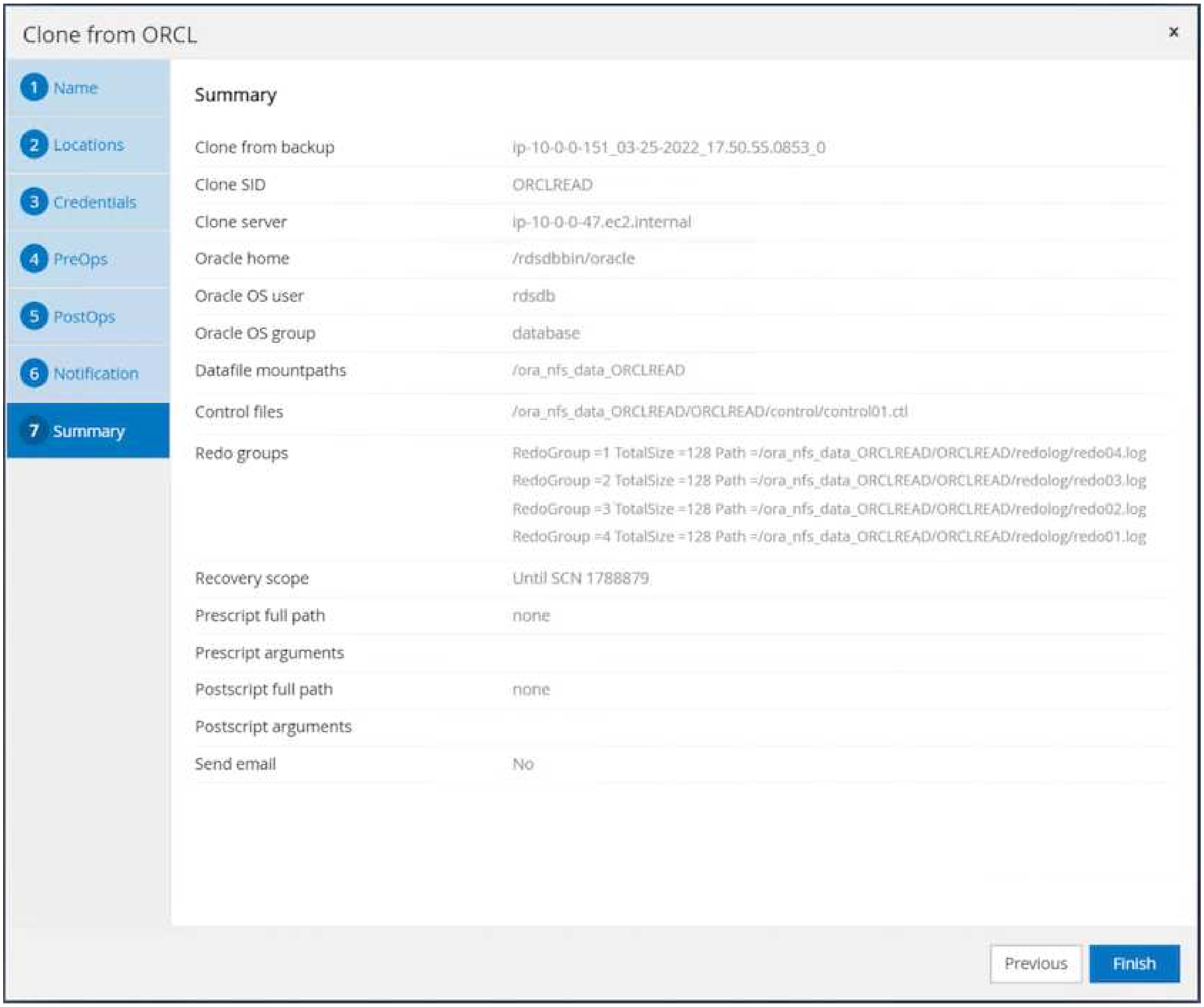

8. Specify a recovery point using either time or the SCN and mounted archive log path.




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.

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

Read the README instruction carefully before attempting setup and failover testing.

Next: Database migration.

## Database migration from on-prem to public cloud

Previous: Database management.

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 a Oracle database backup copy. You must play the archive log to make it current before switching over.

AWS S3 can be used as a staging storage area for database move and migration. See the following high level steps for this method:

1. Provision a new, matching EC2 instance that is comparable with the on-premises instance.
2. Provision equal database volumes from FSx storage and mount the volumes to the EC2 instance.
3. Create a disk-level Oracle backup copy.
4. Move the backup copy to AWS S3 storage.
5. Recreate the Oracle control file and restore and recover the database by pulling data and the archive log from S3 storage.
6. Sync the target Oracle database with the on-premises source database.
7. At switchover, shut down the application and source Oracle database. Copy the last few archive logs and apply them to the target Oracle database to bring it up to date.
8. Start up the target database for user access.
9. Redirect application to the target database to complete the switchover.

#### Migrate on-premises Oracle databases to AWS FSx/EC2 using PDB relocation with maximum availability

This migration approach is best suited to Oracle databases that are already deployed in PDB/CDB multitenant model, and ONTAP storage is not available on-premises. The PDB relocation method utilizes Oracle PDB hot clone technology to move PDBs between a source CDB and a target CDB while minimizing service interruption.

First, create CDB in the AWS FSx/EC2 with sufficient storage to host PDBs to be migrated from on-premises. Multiple on-premises PDBs can be relocated one at a time.

1. If the on-premises database is deployed in a single instance rather than in the multitenant PDB/CDB model, follow the instructions in Converting a single instance non-CDB to a PDB in a multitenant CDB to convert the single instance to multitenant PDB/CDB. Then follow the next step to migrate the converted PDB to CDB in AWS FSx/EC2.
2. If the on-premises database is already deployed in the multitenant PDB/CDB model, follow the instructions in Migrate on-premises Oracle databases to cloud with PDB relocation to perform the migration.

The following video demonstrates how an Oracle database (PDB) can be migrated to FSx/EC2 using PDB relocation with maximum availability.

Migrate on-prem Oracle PDB to AWS CDB with max availability

> Although the instructions in step 1 and 2 are illustrated in the context of Azure public cloud, the procedures are applicable to AWS cloud without any changes.

The NetApp Solutions Automation team provides a migration toolkit that can facilitate Oracle database migration from on-premises to the AWS cloud. Use following command to download the Oracle database migration toolkit for PDB relocation.

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

# Oracle Database Deployment and Protection on Azure NetApp Files

## TR-4954: Oracle Database Deployment and Protection on Azure NetApp Files

Author(s): Allen Cao, Niyaz Mohamed, NetApp

### Overview

Many mission-critical Oracle enterprise databases are still hosted on-premises, and many enterprises are looking to migrate these Oracle databases to a public cloud. Often, these Oracle databases are application centric and thus require user-specific configurations, a capability that is missing from many database-as-a-service public-cloud offerings. Therefore, the current database landscape calls for a public-cloud-based Oracle database solution built from a high-performance, scalable compute and storage service that can accommodate unique requirements. Azure virtual machine compute instances and the Azure NetApp Files storage service might be the missing pieces of this puzzle that you can leverage to build and migrate your mission-critical Oracle database workloads to a public cloud.

### Azure Virtual Machine

Azure virtual machines are one of several types of on-demand, scalable computing resources that Azure offers. Typically, you choose a virtual machine when you need more control over the computing environment than the other choices offer. Azure virtual machines offer a quick and easy way to create a computer with specific configurations required to run your Oracle database, whether it is for compute- or memory-intensive workloads. Virtual machines in an Azure virtual network can easily be connected to your organization's network, for example through a secured VPN tunnel.

### Azure NetApp Files (ANF)

Azure NetApp Files is a fully managed Microsoft service that will take your database workload to the cloud faster and more securely than ever before. It was designed to meet the core requirements of running high-performance workloads such as Oracle databases in the cloud, and it provides performance tiers that reflect the real-world range of IOPS demands, low latency, high availability, high durability, manageability at scale, and fast and efficient backup, recovery, and cloning. These capabilities are possible because Azure NetApp Files is based on physical all-flash NetApp ONTAP systems running within the Azure data center environment. Azure NetApp Files is completely integrated into the Azure DCs and portal, and customers can use the same comfortable graphical interface and APIs for creating and managing shared files as with any other Azure object. With Azure NetApp file, you can unlock the full capabilities of Azure without extra risk, cost, or time and trust the only enterprise file service native to Azure.

### Conclusion

This documentation describes in detail how to deploy, configure, and protect an Oracle database with an Azure virtual machine and Azure NetApp Files storage service that delivers performance and durability similar to an on-premises system. For best-practices guidance, see TR-4780 Oracle Databases on Microsoft Azure. More importantly, NetApp also provides automation toolkits that automate most of the tasks that are required for the deployment, configuration, data protection, migration, and management of your Oracle database workload in the Azure public cloud. The automation toolkits are available for download at NetApp public GitHub site: NetApp-Automation.

Next: Solutions architecture.

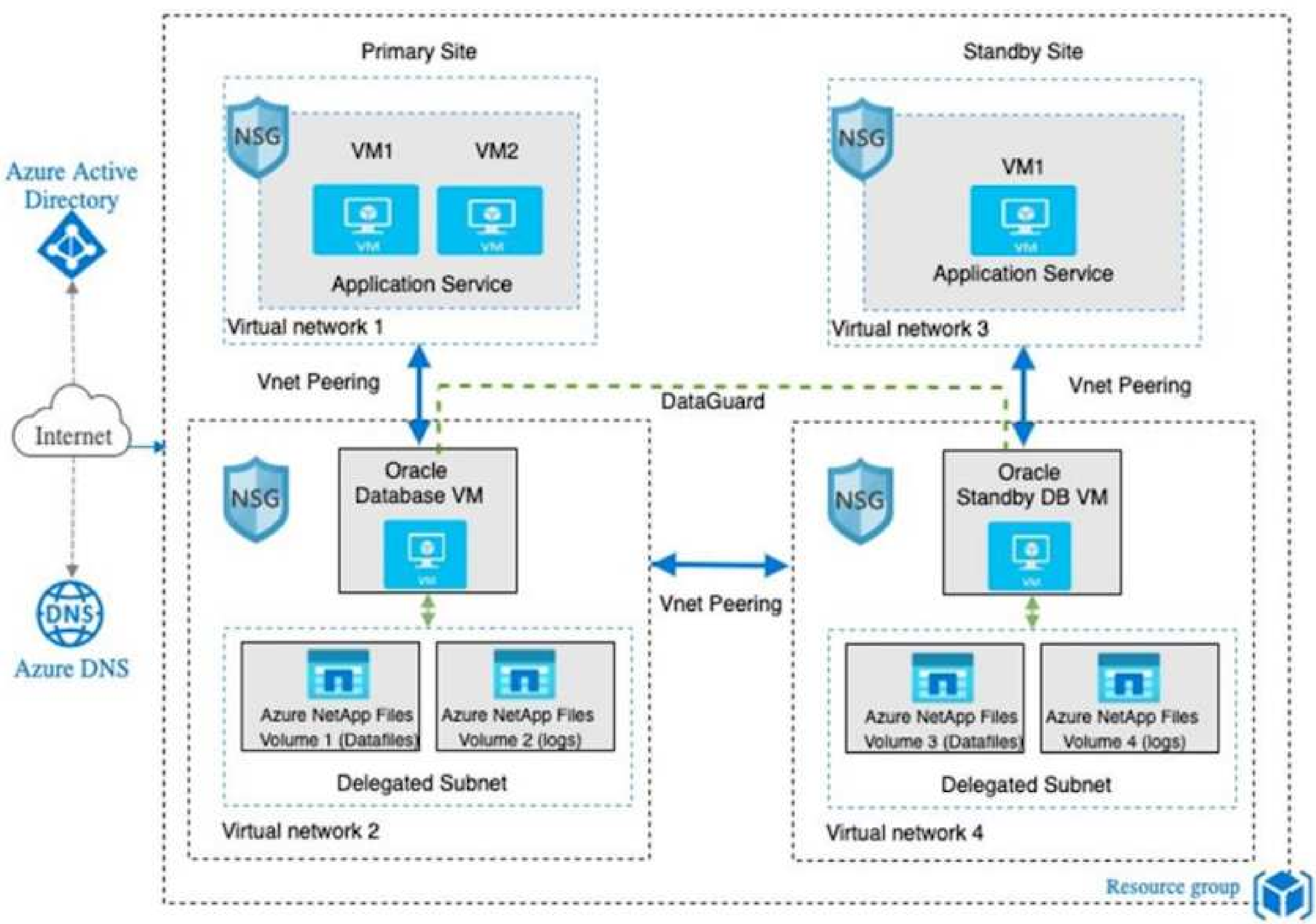
## Solution Architecture

Previous: Introduction.

### The following architecture diagram illustrates a highly available Oracle database deployment on Azure VM instances and the Azure NetApp Files storage.

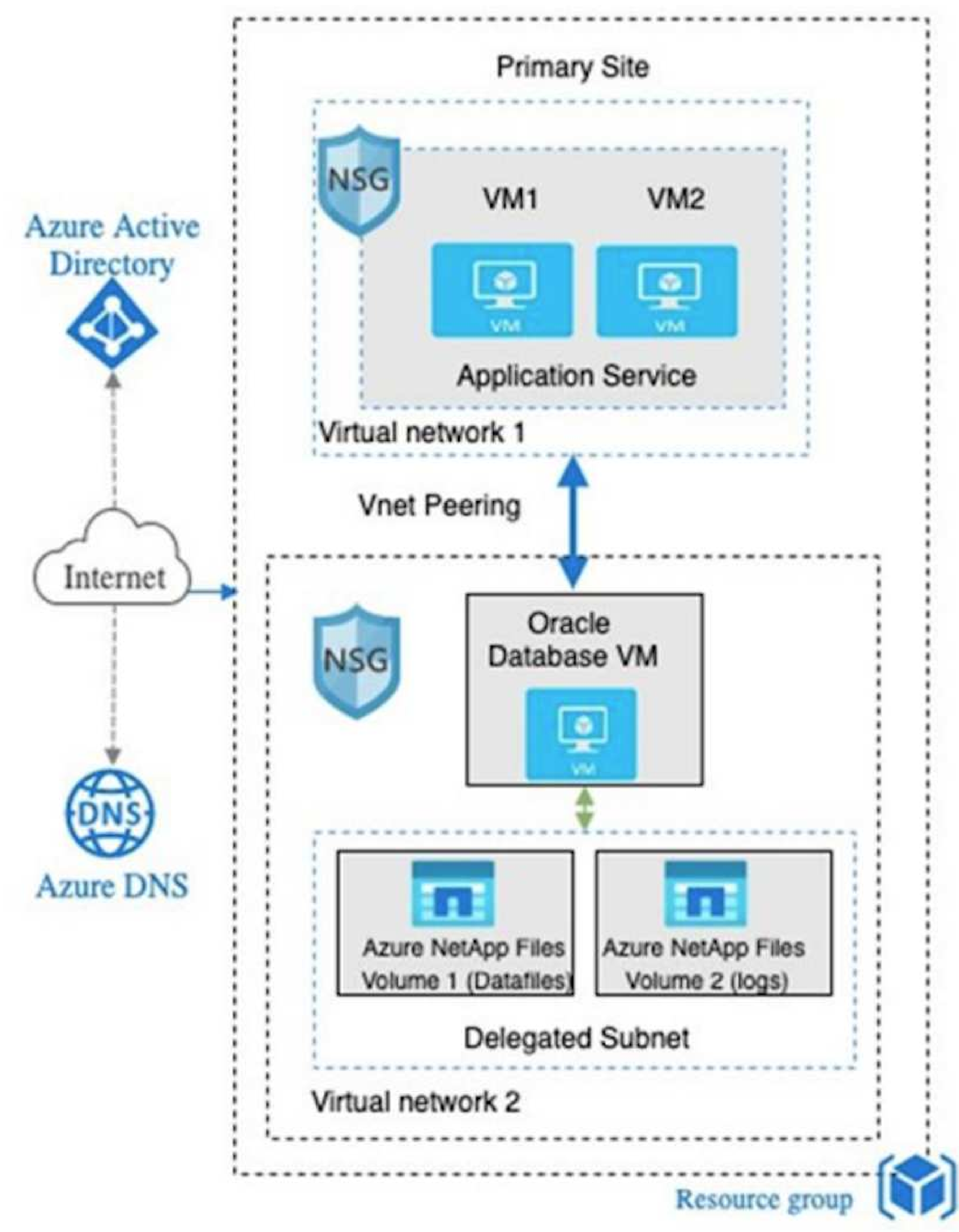
Within the environment, the Oracle compute instance is deployed via an Azure services VM console. There are multiple Azure instance types available from the console. NetApp recommends deploying a database-oriented Azure VM instance that meets your expected workload.

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.

Next: Factors to consider.

## Factors to consider for Oracle database deployment

Previous: Solution architecture.

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

Azure NetApp Files supports three service levels: Ultra, Premium, and Standard.

- **Ultra storage.** This tier provides up to 128MiBps of throughput per 1TiB of volume quota assigned.
- **Premium storage.** This tier provides up to 64MiBps of throughput per 1TiB of volume quota assigned.
- **Standard storage.** This tier provides up to 16MiBps of throughput per 1TiB of volume quota assigned.

#### 2. Capacity pool and quality of service

Each of the desired service levels has an associated cost for provisioned capacity and includes a quality-of-service (QoS) level that defines the overall maximum throughput for provisioned space.

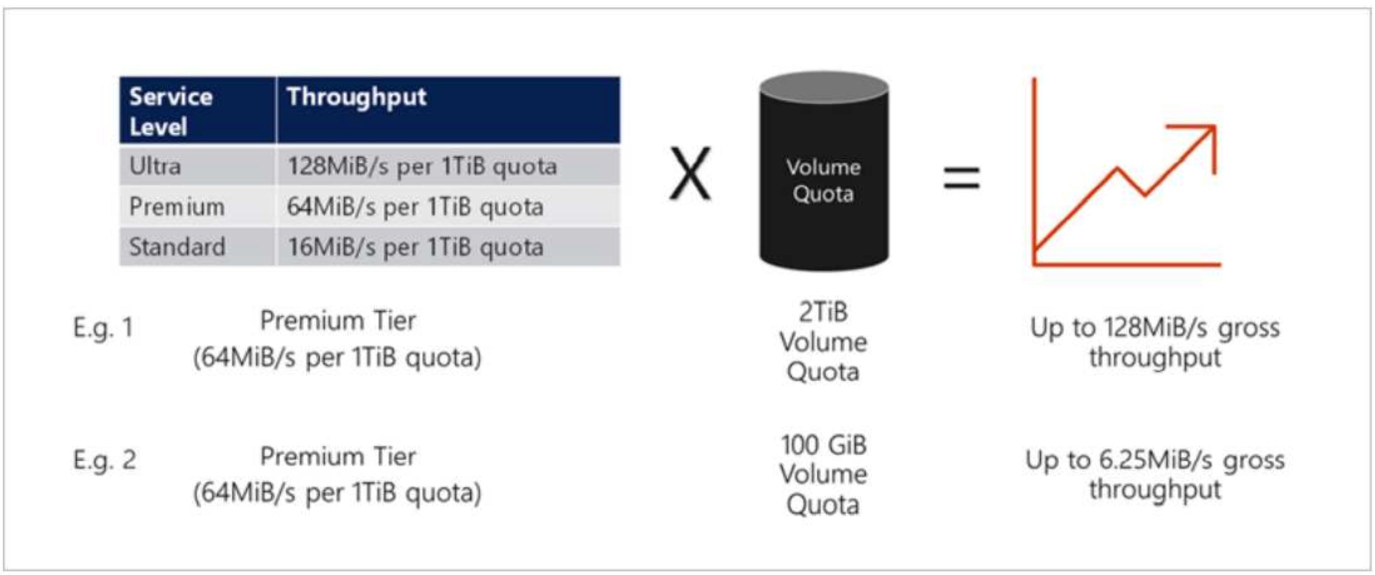
For example, a 10TiB-provisioned single-capacity pool with the premium service level provides an overall available throughput for all volumes in this capacity pool of 10x 64MBps, so 640MBps with 40,000 (16K) IOPs or 80,000 (8K) IOPs.

The minimum capacity pool size is 4TiB. You can change the size of a capacity pool in 1TiB increments in response to changes in your workload requirements to manage storage needs and costs.

#### 3. Calculate the service level at a database volume

The throughput limit for an Oracle database volume is determined by a combination of the following factors: The service level of the capacity pool to which the volume belongs and The quota assigned to the volume.

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




In example 1, a volume from a capacity pool with the Premium storage tier that is assigned 2TiB of quota is assigned a throughput limit of 128MiBps (2TiB * 64MiBps). This scenario applies regardless of the capacity pool size or the actual volume consumption.

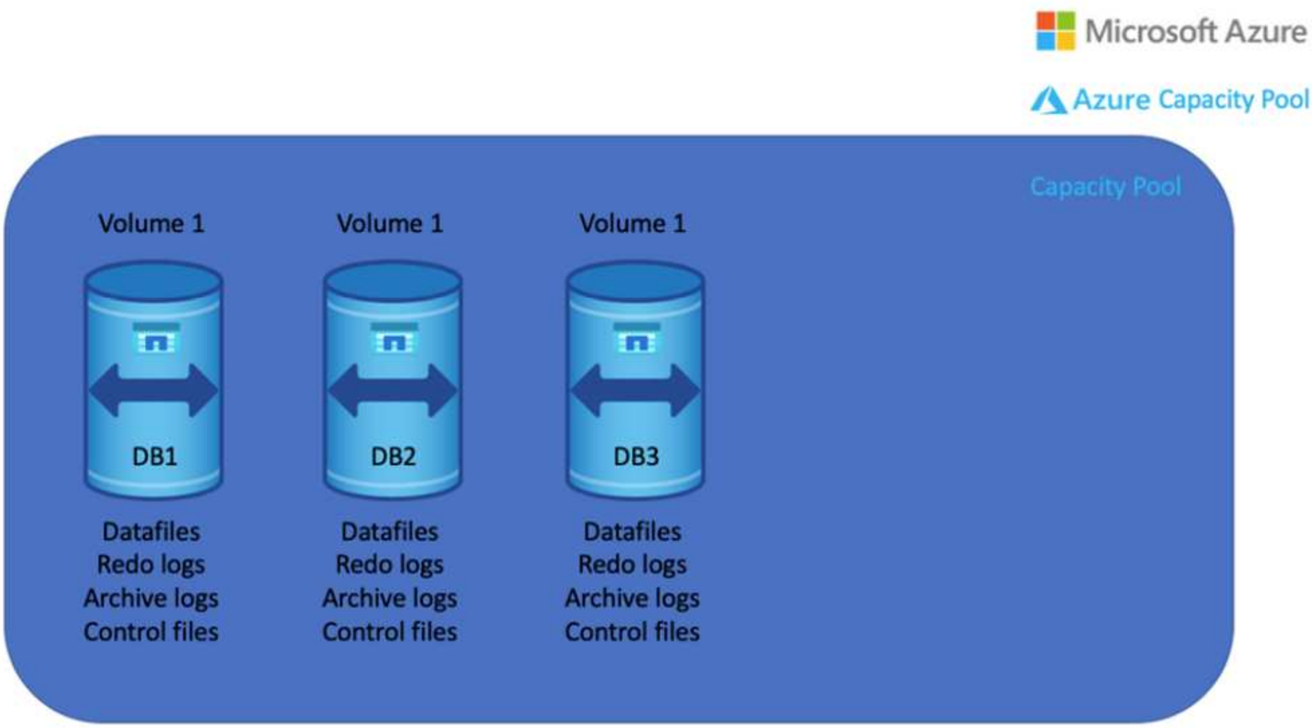
In example 2, a volume from a capacity pool with the Premium storage tier that is assigned 100GiB of quota is assigned a throughput limit of 6.25MiBps (0.09765625TiB * 64MiBps). This scenario applies regardless of the capacity pool size or the actual volume consumption.

Please note that the minimum volume size is 100GiB.

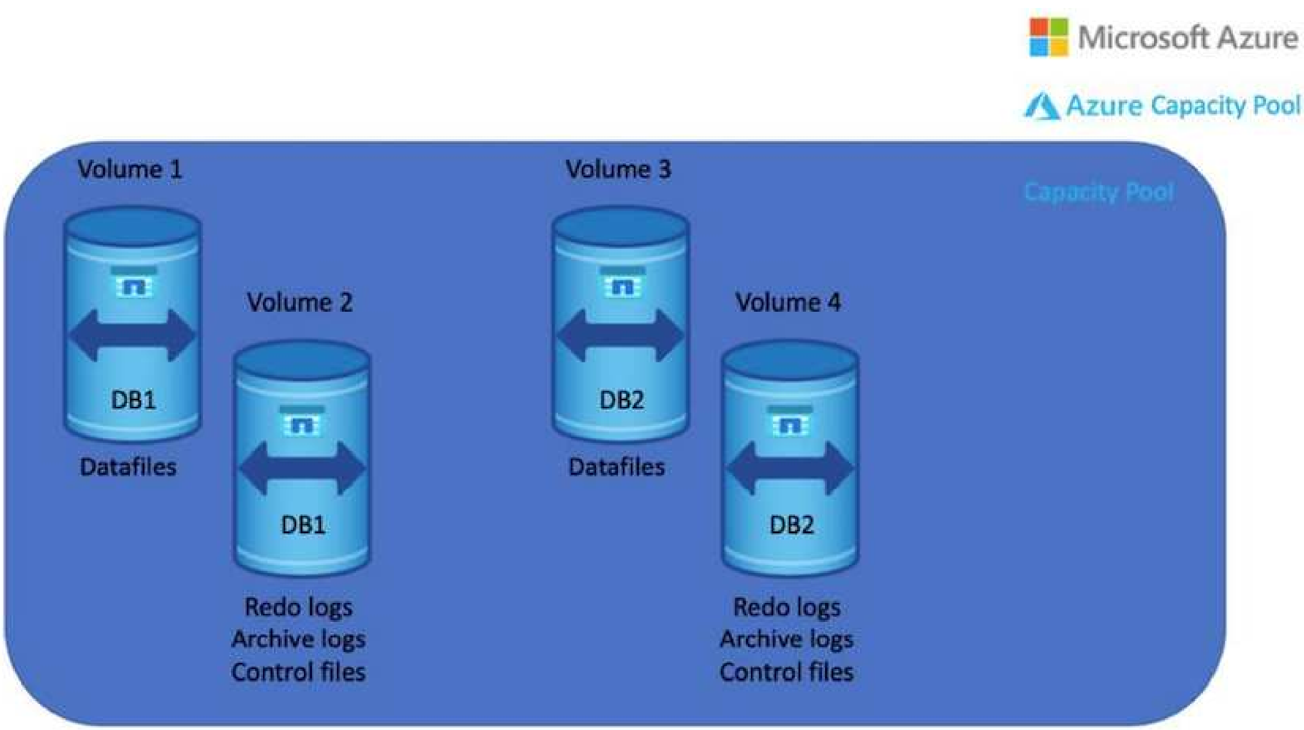
### Storage layout and settings

NetApp recommends the following storage layout:

- For small databases, using single volume layout for all Oracle files.




- For large databases, the recommended volume layout is multiple volumes: one for Oracle data and a duplicate control file and one for the Oracle active log, archived log, and control file. NetApp highly recommends allocating a volume for the Oracle binary instead of the local drive so that the database can be relocated to a new host and quickly restored.

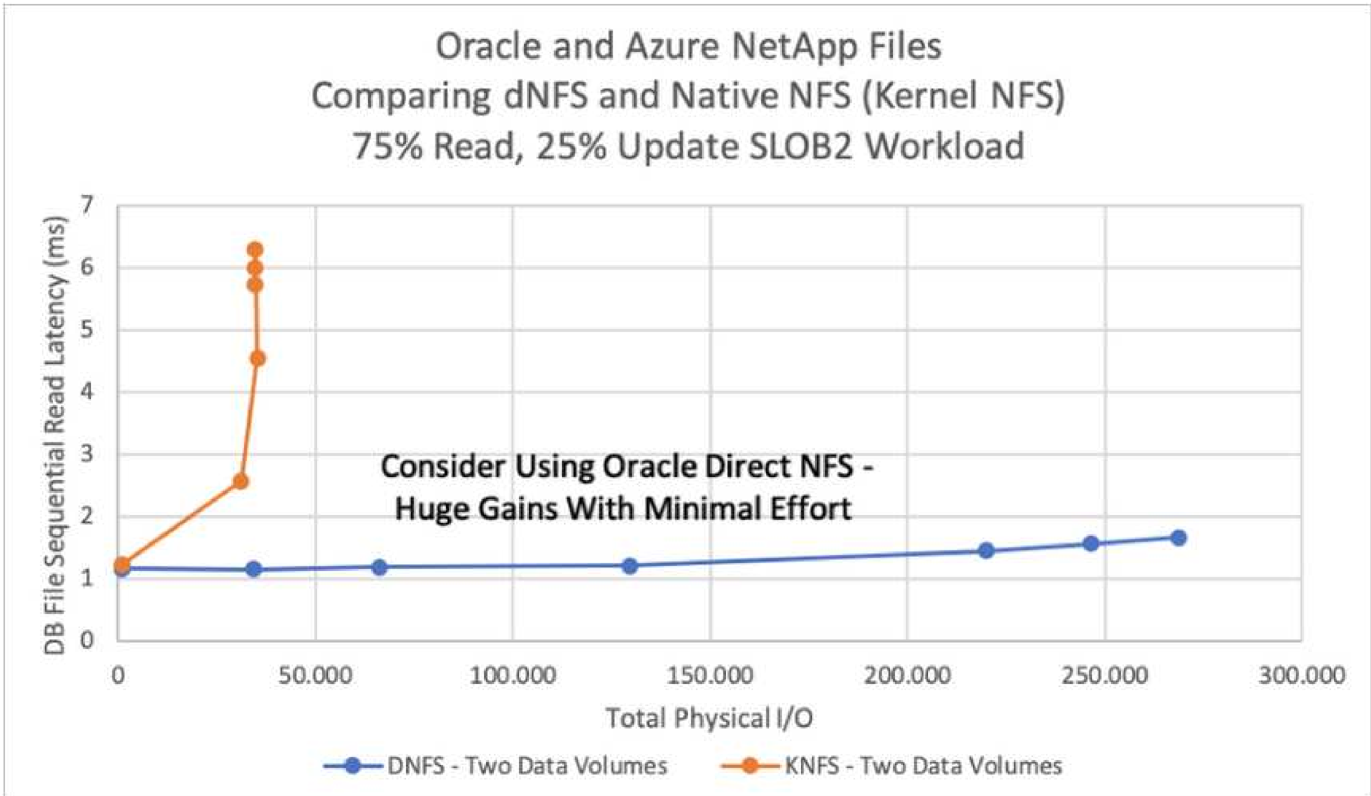


### NFS configuration

Linux, the most common operating system, includes native NFS capabilities. Oracle offers a direct NFS (dNFS) client natively integrated into Oracle. Oracle dNFS bypasses the OS cache and enables parallel processing to improve database performance. Oracle has supported NFSv3 for over 20 years, and NFSv4 is supported with Oracle 12.1.0.2 and later.

By using dNFS (available since Oracle 11g), an Oracle database running on an Azure Virtual Machine can drive significantly more I/O than the native NFS client. Automated Oracle deployment using the NetApp automation toolkit automatically configures dNFS on NFSv3.

The following diagram demonstrates the SLOB benchmark on Azure NetApp Files with Oracle dNFS.




Other factors to consider:

- TCP slot tables are the NFS equivalent of host-bus-adapter (HBA) queue depth. These tables control the number of NFS operations that can be outstanding at any one time. The default value is usually 16, which is far too low for optimum performance. The opposite problem occurs on newer Linux kernels, which can automatically increase the TCP slot table limit to a level that saturates the NFS server with requests.

  For optimum performance and to prevent performance problems, adjust the kernel parameters that control TCP slot tables to 128.

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

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

| File Type | Mount Options |
| --- | --- |
| • Control files<br>• Data files<br>• Redo logs | `rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536` |
| • ORACLE_HOME<br>• ORACLE_BASE | `rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536` |

> 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.

Next: Deployment procedures.

## Step-by-Step Oracle deployment procedures on Azure VM and Azure NetApp Files

Previous: Factors to consider.

### 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

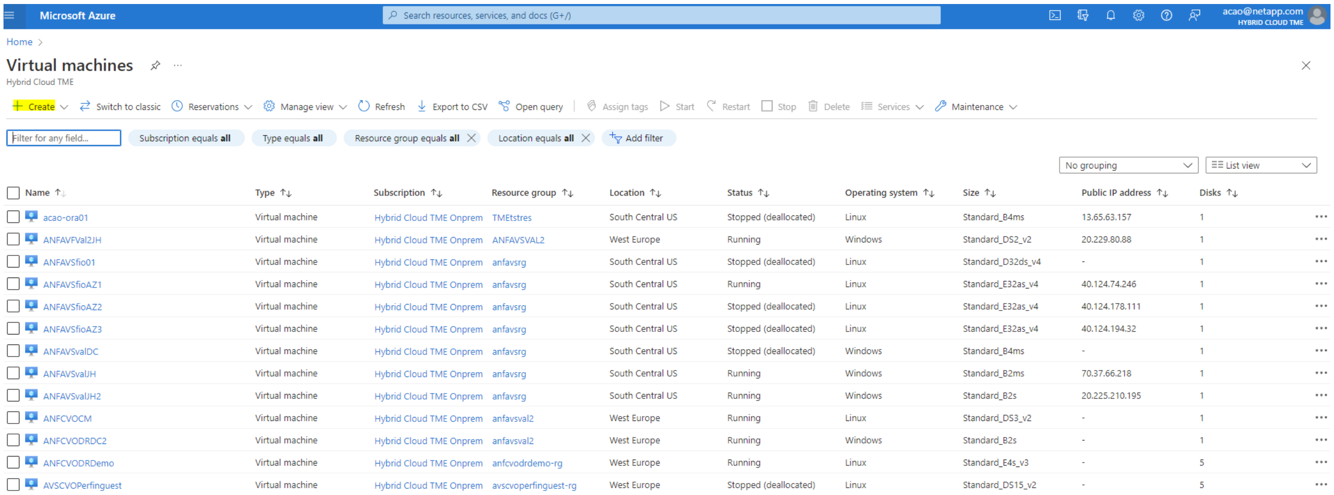
In this deployment scenario, an Azure VM OS is provisioned using regular Azure storage, but ANF volumes are provisioned to run Oracle database via NFS. First, you need to create a NetApp storage account and a capacity pool to host the storage volumes.

See Set up Azure NetApp Files and create an NFS volume to set up an ANF capacity pool.

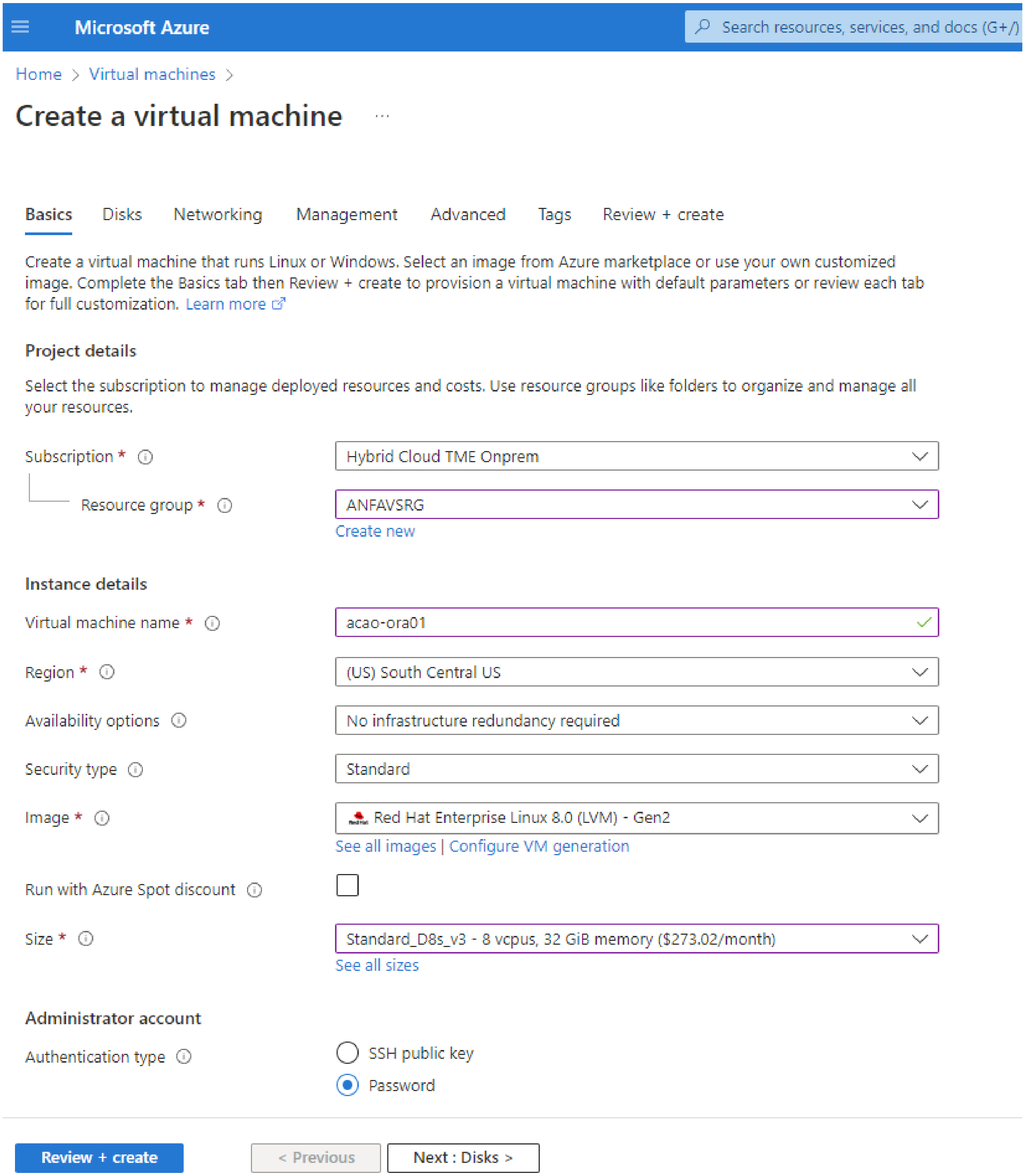
##### 3. Provision Azure VM for Oracle

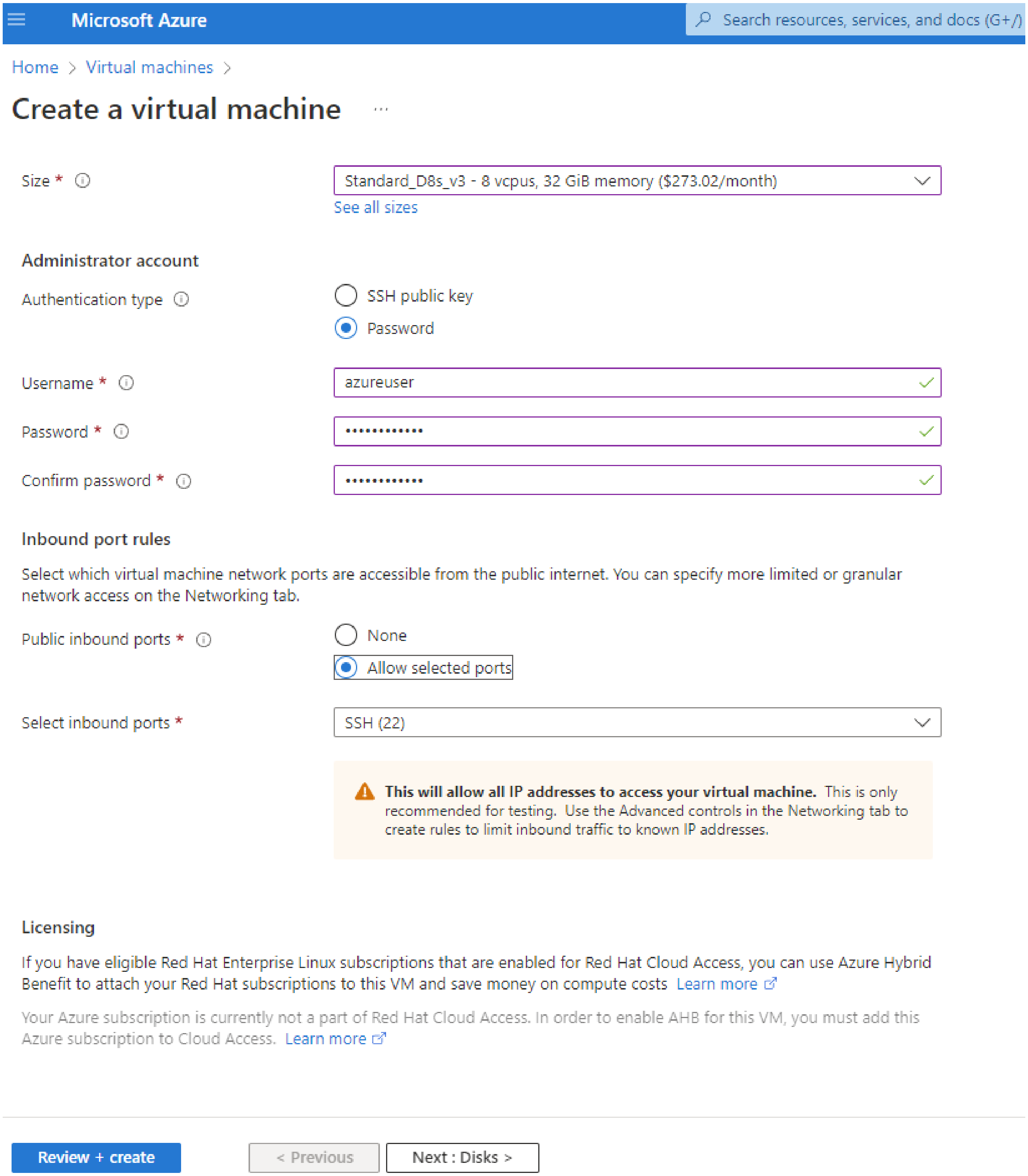
Based on your workload, determine what type of Azure VM you need and the size of the VM vCPU and RAM to deploy for Oracle. Then, from the Azure console, click the VM icon to launch the VM deployment workflow.

1. From the Azure VM page, click **Create** and then choose **Azure virtual machine**.

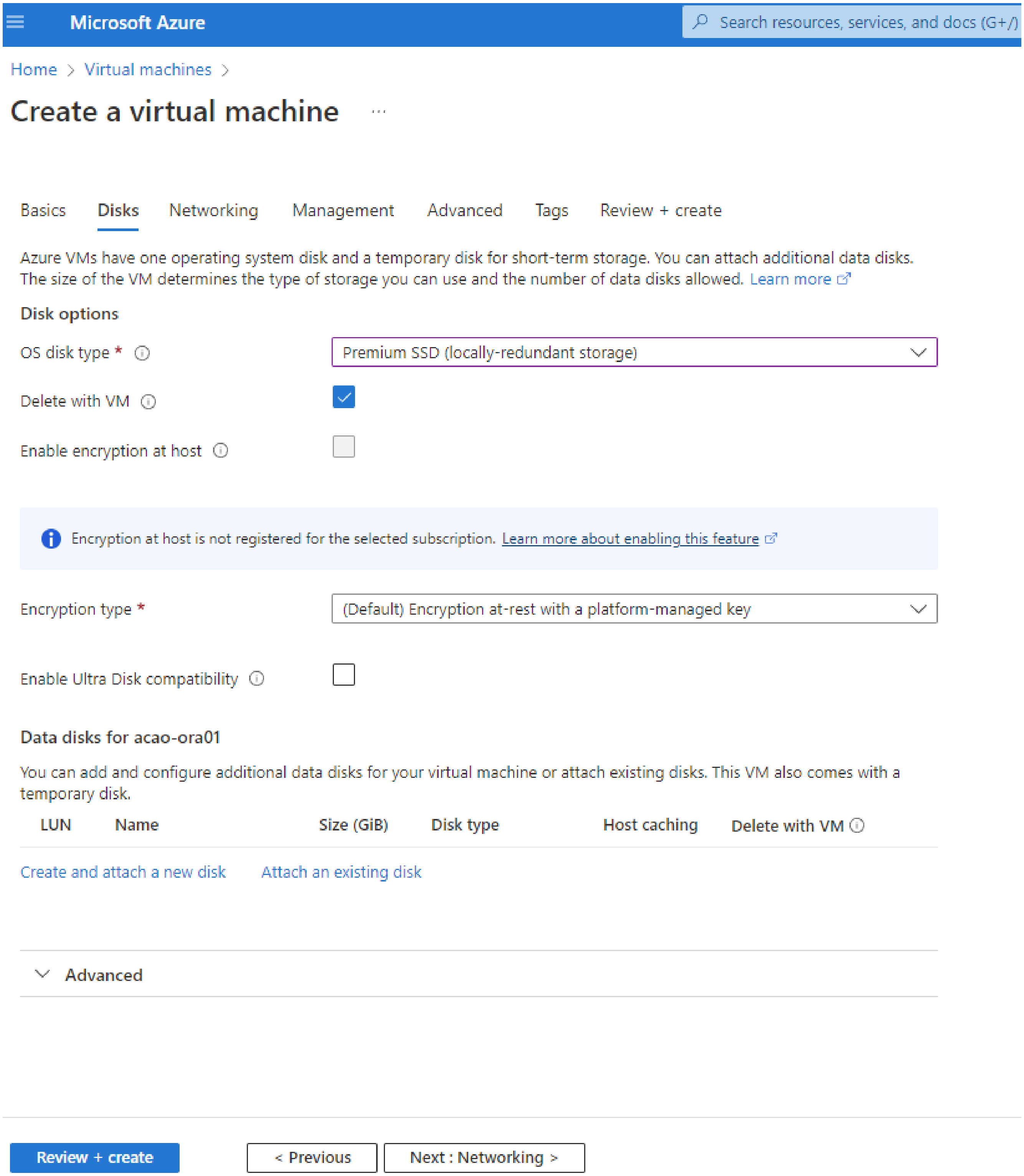


2. Choose the subscription ID for the deployment, and then choose the resource group, region, host name, VM image, size, and authentication method. Go to the Disk page.

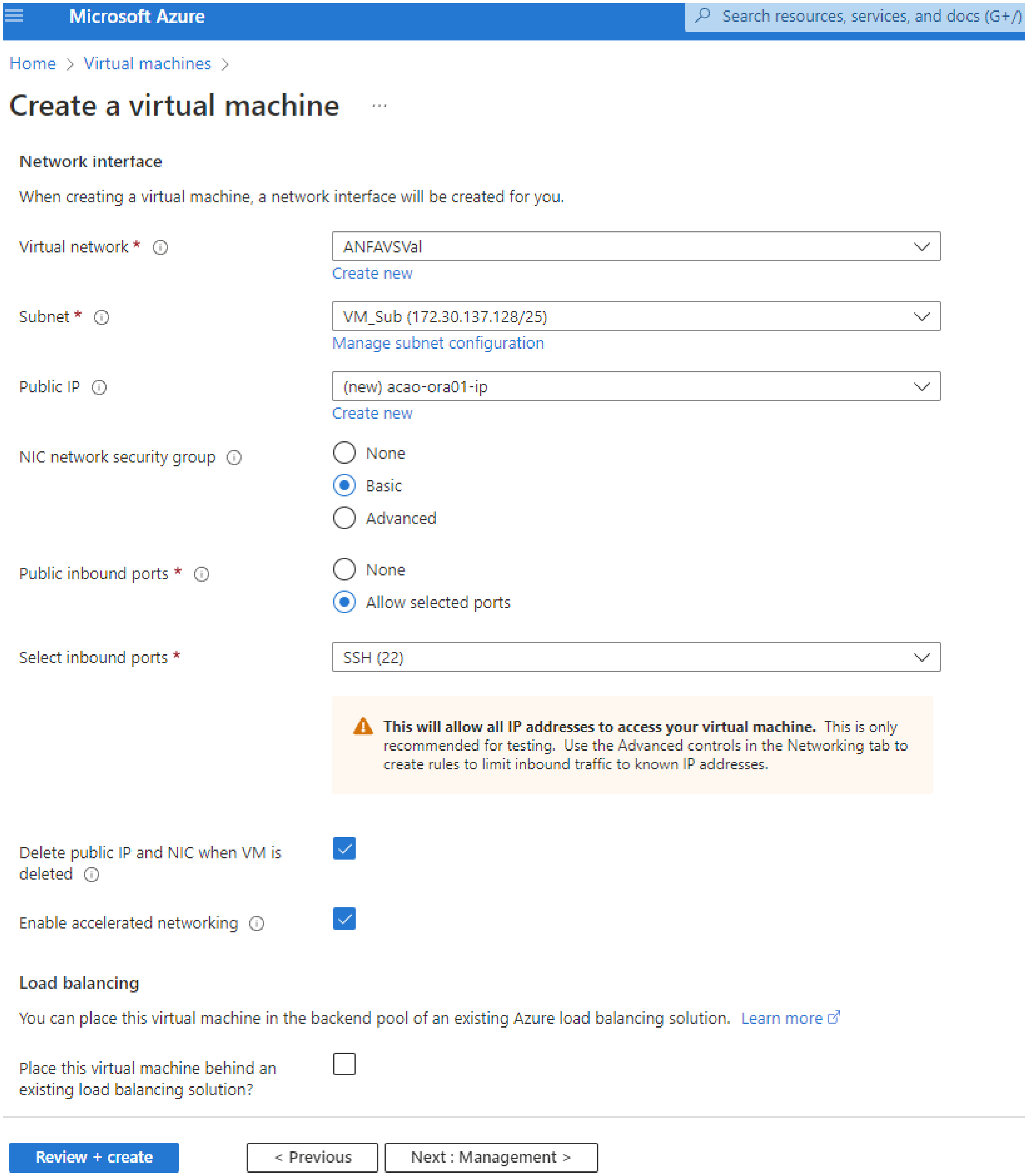

Microsoft Azure
Search resources, services, and docs (G+/)
Home > Virtual machines >
Create a virtual machine ...
Basics
Disks
Networking
Management
Advanced
Tags
Review + create
Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. Learn more
Project details
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.
Subscription *
Hybrid Cloud TME Onprem
Resource group *
ANFAVSRG
Create new
Instance details
Virtual machine name *
acao-ora01
Region *
(US) South Central US
Availability options
No infrastructure redundancy required
Security type
Standard
Image *
Red Hat Enterprise Linux 8.0 (LVM) - Gen2
See all images | Configure VM generation
Run with Azure Spot discount
Size *
Standard_D8s_v3 - 8 vcpus, 32 GiB memory ($273.02/month)
See all sizes
Administrator account
Authentication type
SSH public key
Password
Review + create
< Previous
Next : Disks >



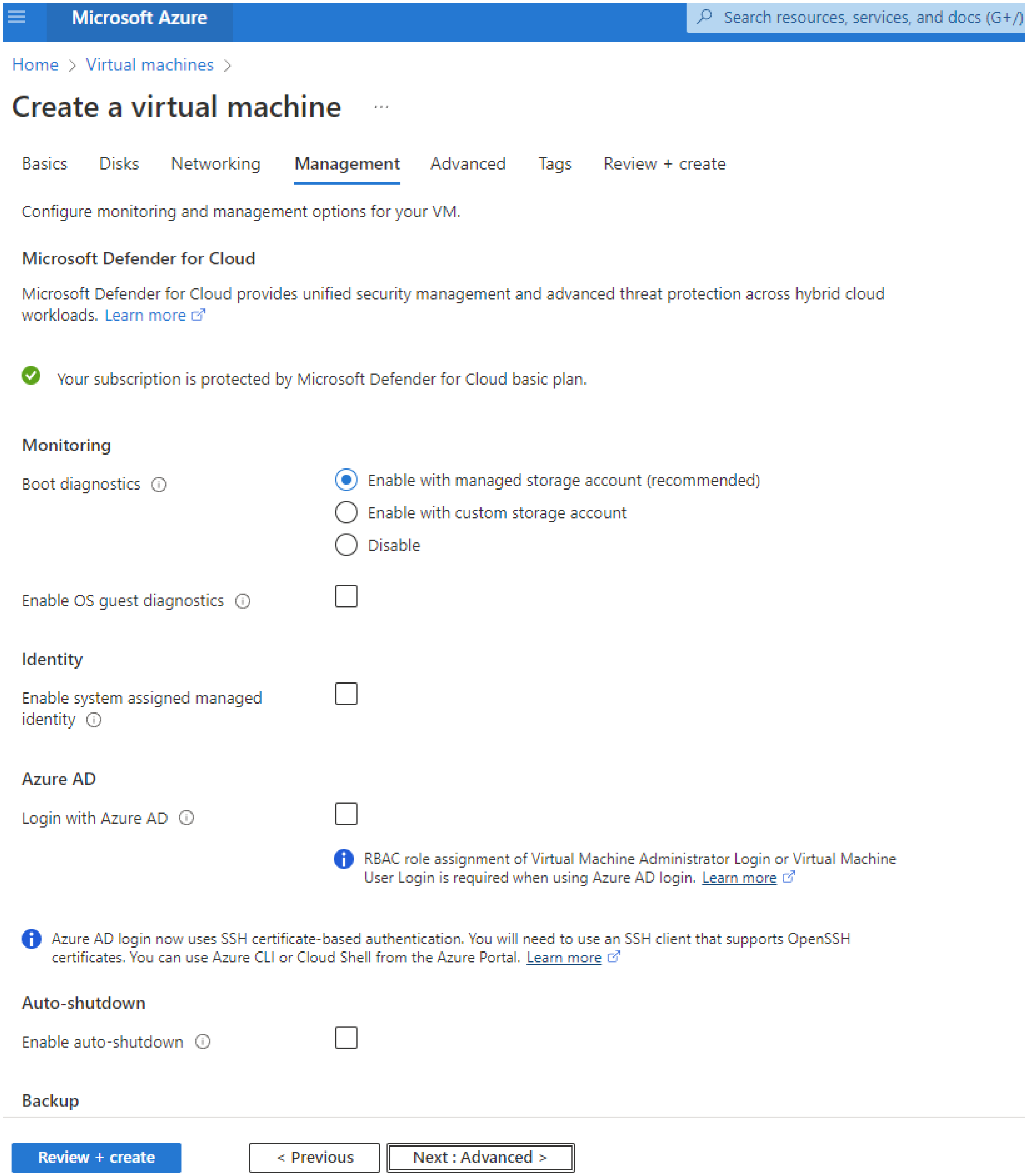

3. Choose **premium SSD** for OS local redundancy and leave the data disk blank because the data disks are mounted from ANF storage. Go to the Networking page.



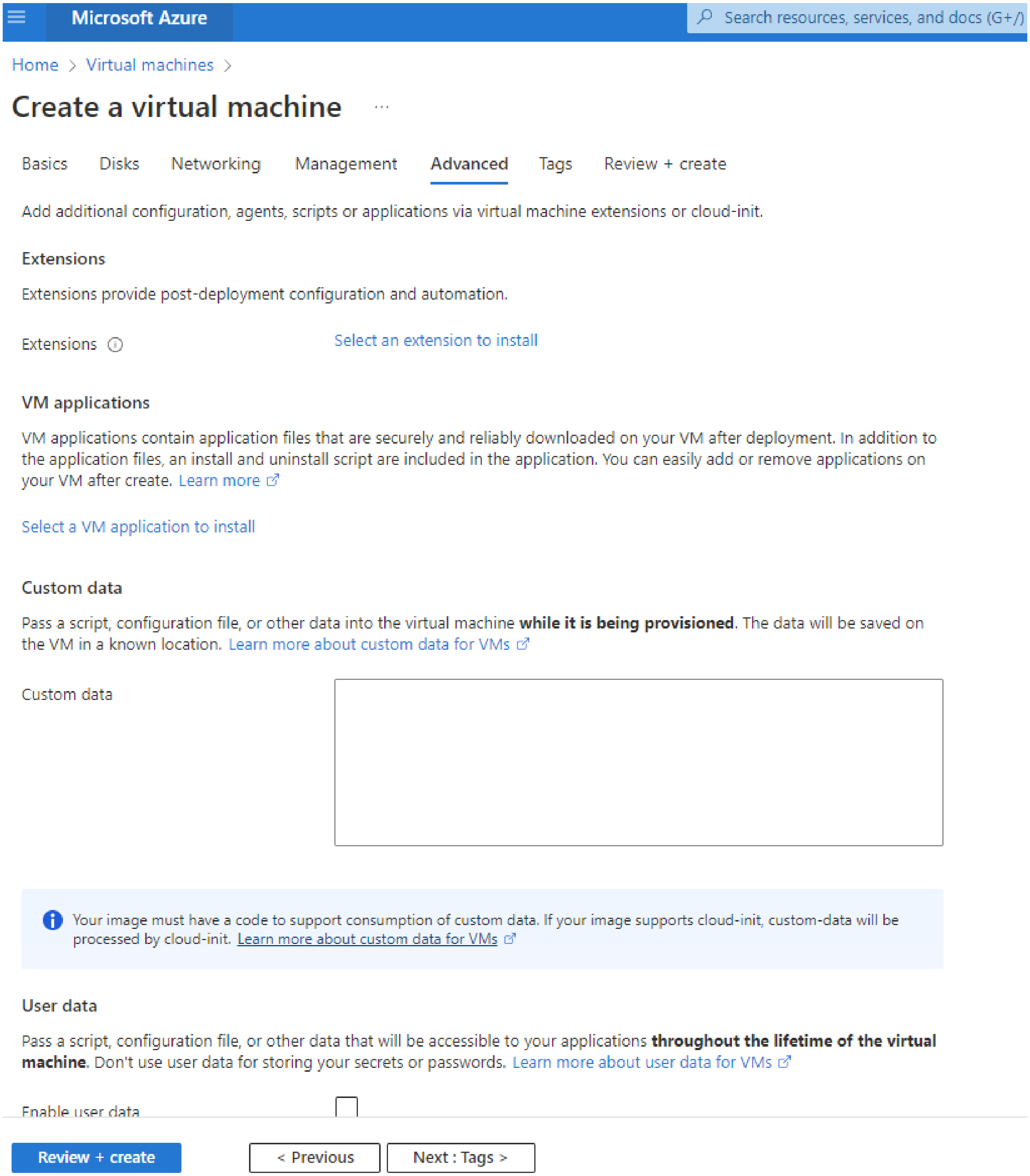

4. Choose the VNet and subnet. Allocate a public IP for external VM access. Then go to the Management page.



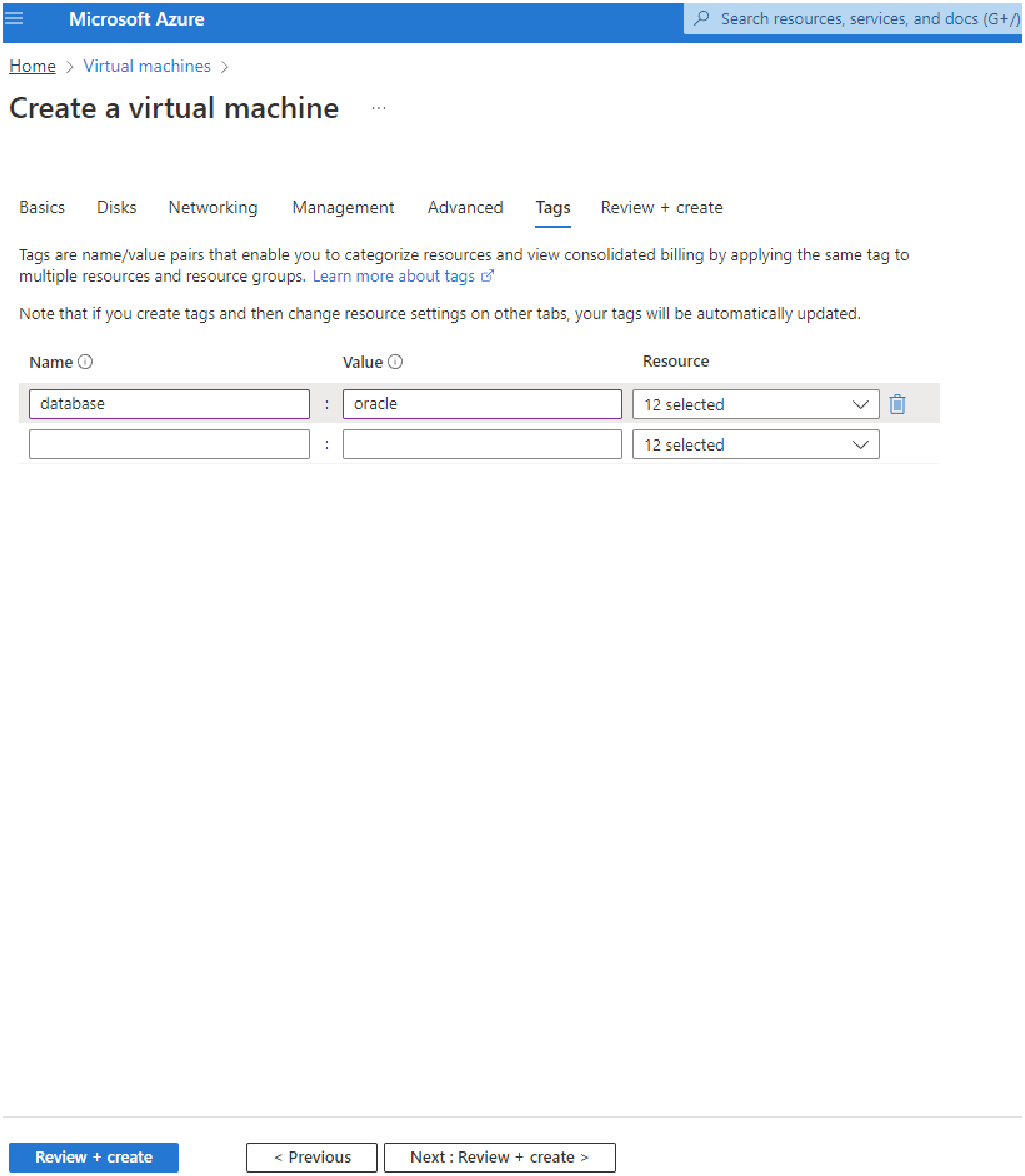

5. Keep all defaults for Management and move to the Advanced page.




6. Keep all defaults for the Advanced page unless you need to customize a VM after deployment with custom scripts. Then go to Tags page.




7. Add a tag for the VM if desired. Then, go to the Review + create page.

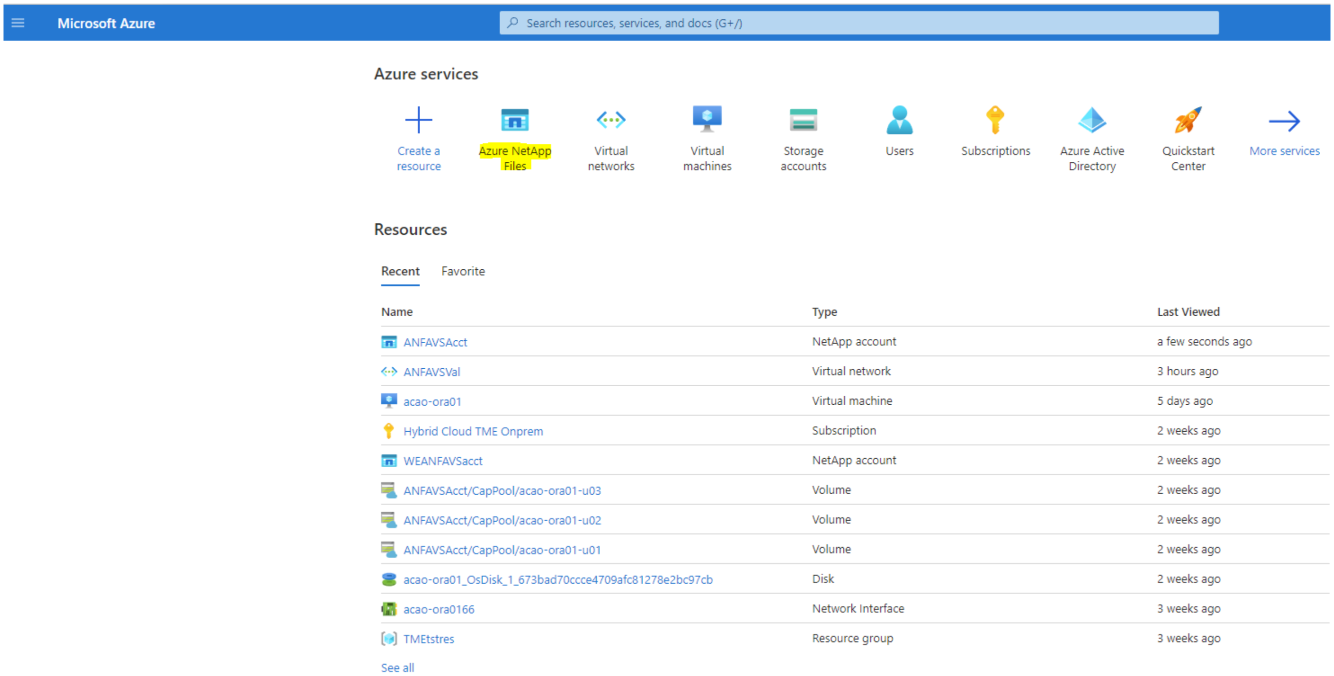



8. The deployment workflow runs a validation on the configuration, and, if the validation passes, click **Create** to create the VM.

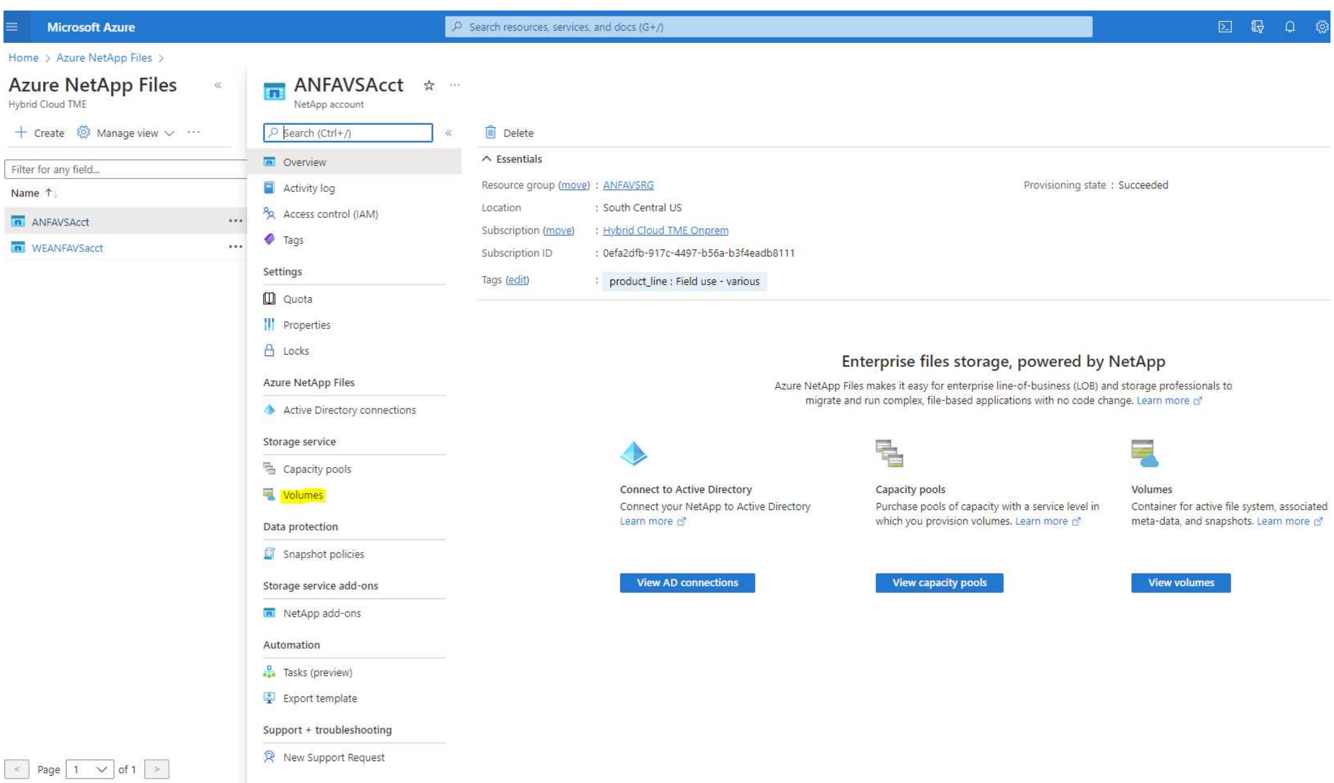
#### 4. Provision ANF database volumes for Oracle

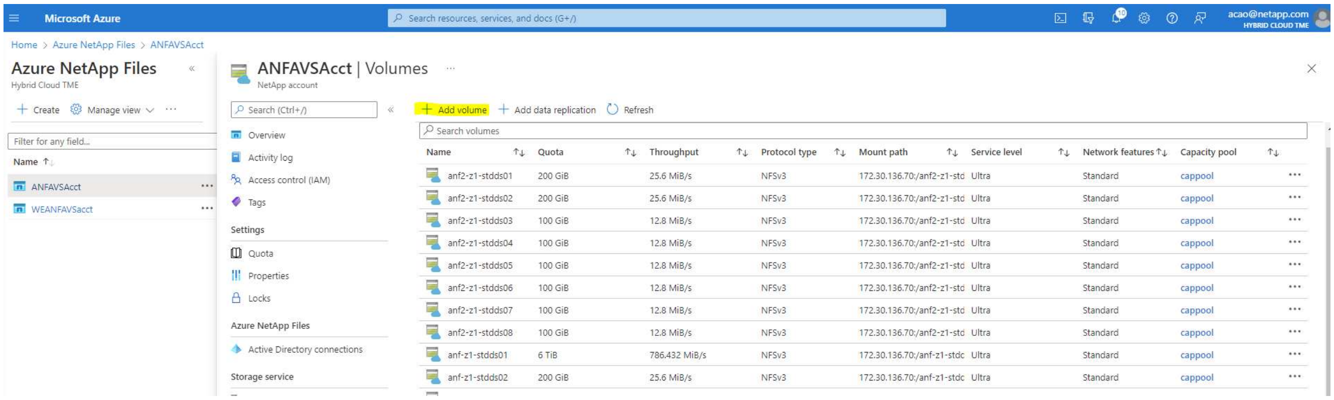
You must create three NFS volumes for an ANF capacity pool for the Oracle binary, data, and log volumes respectively.

1. From the Azure console, under the list of Azure services, click Azure NetApp Files to open a volume creation workflow. If you have more than one ANF storage account, click the account that you would like to provision volumes from.

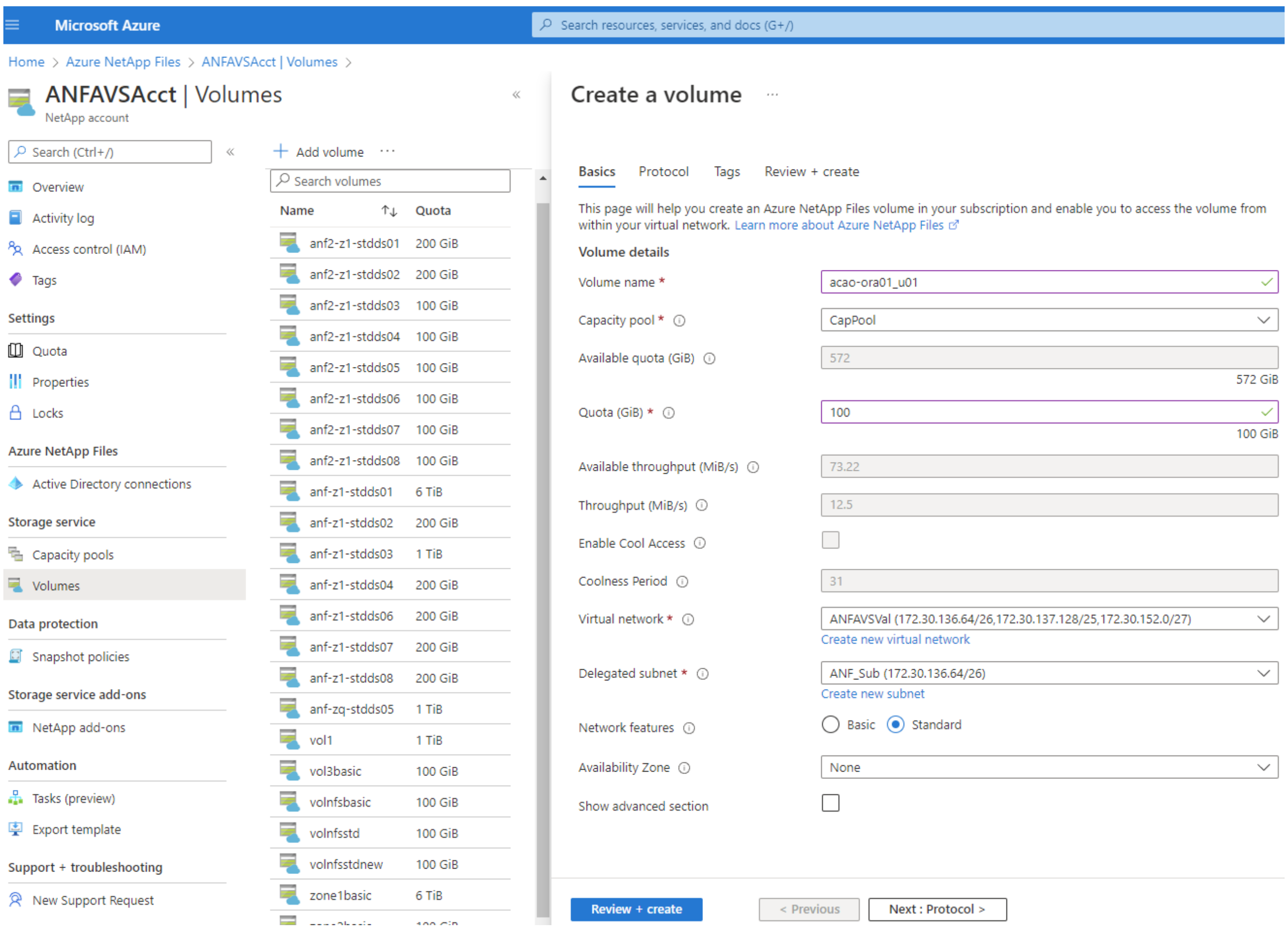



2. Under your NetApp storage account, click **Volumes**, and then **Add volume** to create new Oracle volumes.

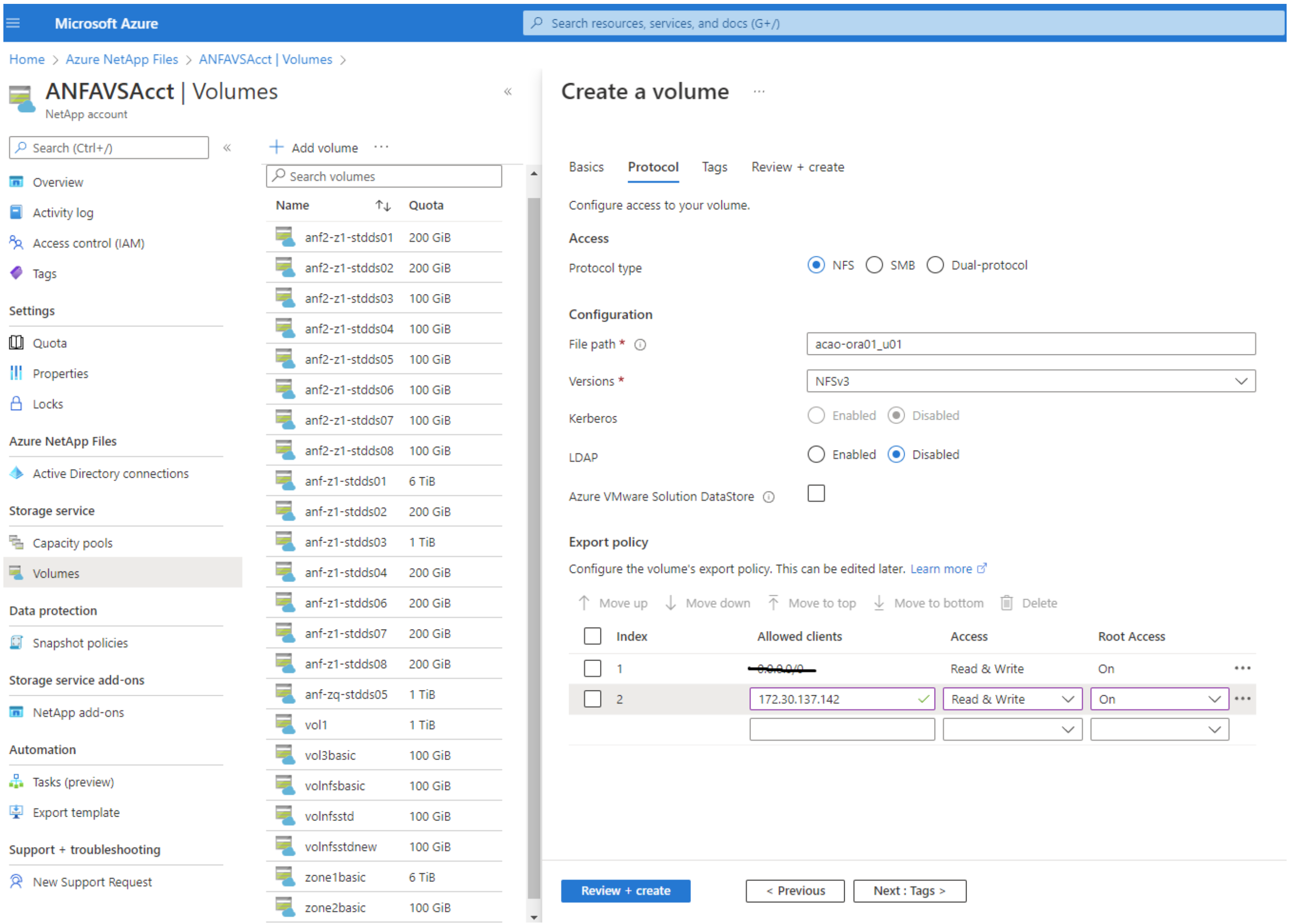




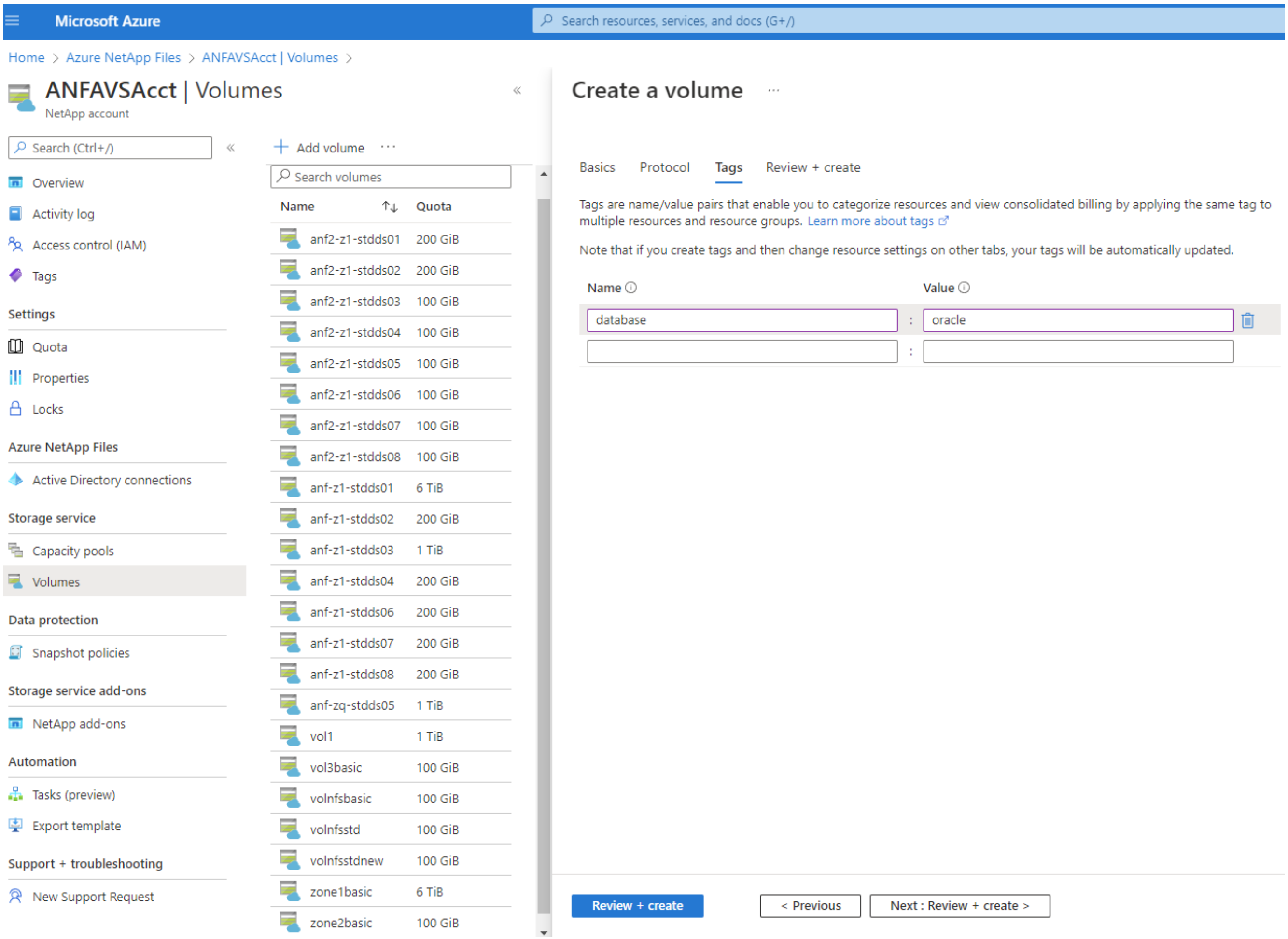

3. As a good practice, identify Oracle volumes with the VM hostname as a prefix and then followed by the mount point on the host, such as u01 for Oracle binary, u02 for Oracle data, and u03 for Oracle log. Choose the same VNet for the volume as for the VM. Click **Next: Protocol>**.



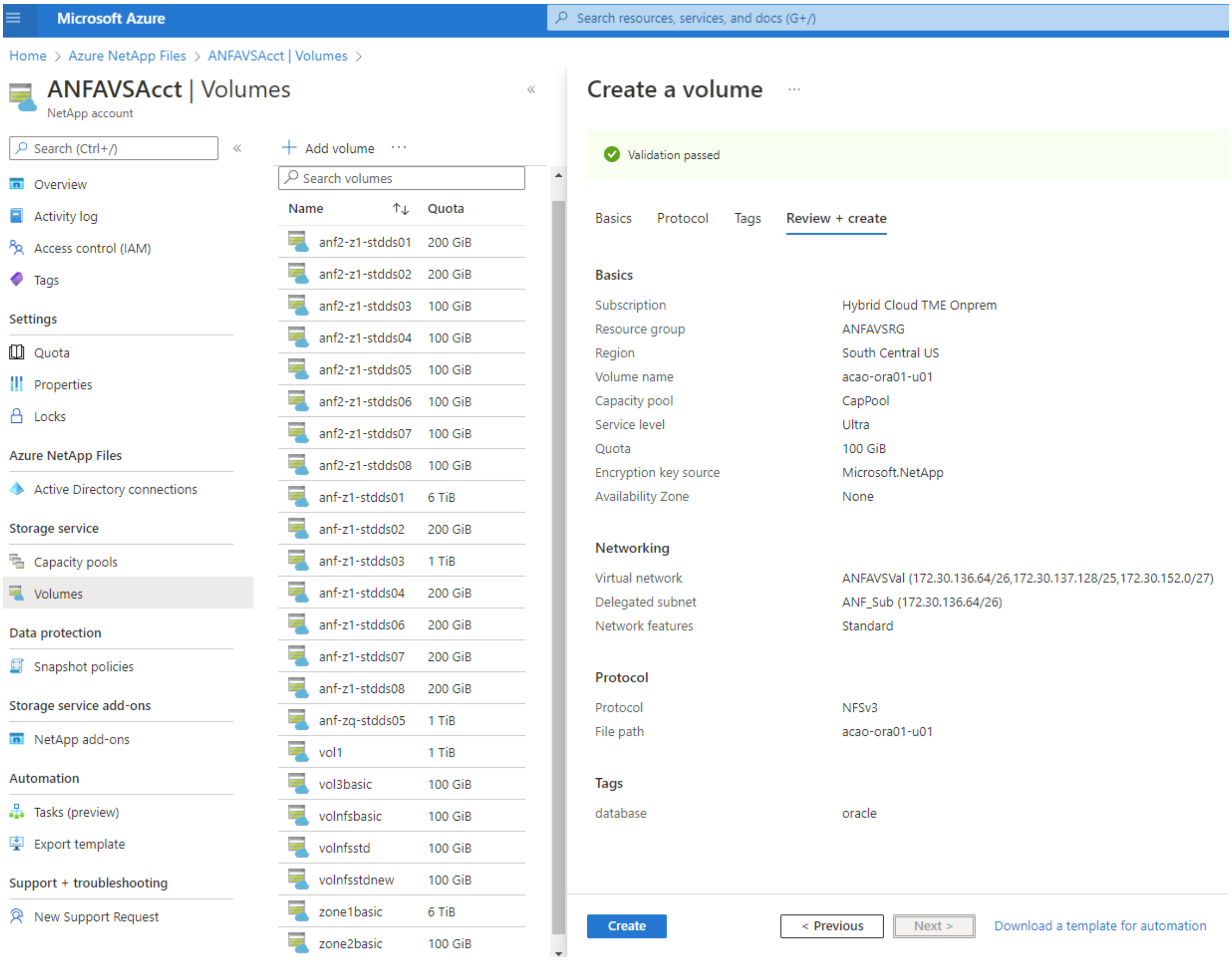

4. Choose the NFS protocol, add the Oracle host IP address to the allowed client, and remove the default policy that allows all IP addresses 0.0.0.0/0. Then click **Next: Tags>**.




5. Add a volume tag if desired. Then click **Review + Create>**.




6. If the validation passes, click **Create** to create the volume.




### Install and configure Oracle on Azure VM with ANF

The NetApp solutions team has created many Ansible-based automation toolkits to help you deploy Oracle in Azure smoothly. Follow these steps to deploy Oracle on an Azure VM.

#### Set up an Ansible controller

If you have not set up an Ansible controller, see NetApp Solution Automation, which has detailed instructions on how to setup an Ansible controller.

#### Obtain Oracle deployment automation toolkit

Clone a copy of the Oracle deployment toolkit in your home directory under the user ID that you use to log into the Ansible controller.

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

#### Execute the toolkit with your configuration

See the CLI deployment Oracle 19c Database to execute the playbook with the CLI. You can ignore the ONTAP portion of the variables configuration in the global VARS file when you create database volumes from the Azure console rather than the CLI.

> The toolkit default deploys Oracle 19c with RU 19.8. It can be easily adapted for any other patch level with minor default configuration changes. Also default seed-database active log files are deployed into the data volume. If you need active log files on the log volume, it should be relocated after initial deployment. Reach out to the NetApp Solution team for help if needed.

### Set up AzAcSnap backup tool for app-consistent snapshots for Oracle

The Azure Application-Consistent Snapshot tool (AzAcSnap) is a command-line tool that enables data protection for third-party databases by handling all the orchestration required to put them into an application-consistent state before taking a storage snapshot. It then returns these databases to an operational state. NetApp recommends installing the tool on the database server host. See the following installation and configuration procedures.

#### Install AzAcSnap tool

1. Get the most recent version of the 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. It's 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.

```
mkstore -wrl $TNS_ADMIN/.oracle_wallet/ -createCredential AZACSNAP
azacsnap AzPasswd1
```

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.

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

6. Create the `sqlnet.ora` file.

```
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.

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

The expected output from the command:

```
[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/netAppA
ccounts/.../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
```

```
ccounts/.../capacityPools/Premium/volumes/...): /subscriptions/0efa2dfb-
917c-4497-b56a-
b3f4eadb8111/resourceGroups/ANFAVSRG/providers/Microsoft.NetApp/netAppAc
counts/ANFAVSAcct/capacityPools/CapPool/volumes/acao-ora01-u02
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]: n
--- OTHER Volumes are snapshot immediately without preparing any
application for snapshot ---
Add Azure NetApp Files resource to OTHER 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
ccounts/.../capacityPools/Premium/volumes/...): /subscriptions/0efa2dfb-
917c-4497-b56a-
b3f4eadb8111/resourceGroups/ANFAVSRG/providers/Microsoft.NetApp/netAppAc
counts/ANFAVSAcct/capacityPools/CapPool/volumes/acao-ora01-u03
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 OTHER Volume section of Database
configuration? (y/n) [n]: n

=== Azure Managed Disk details ===
Are you using Azure Managed Disks for the database? (y/n) [n]: n

=== Azure Large Instance (Bare Metal) Storage details ===
Are you using Azure Large Instance (Bare Metal) for the database? (y/n)
[n]: n

Enter the database type to add, 'hana', 'oracle', or 'exit' (for no
database): exit

Editing configuration complete, writing output to 'azacsnap.json'.
```

4. As the azacsnap Linux user, run the azacsnap test command for an Oracle backup.

```
cd ~/bin
azacsnap -c test --test oracle --configfile azacsnap.json
```

The expected output:

```
[azacsnap@acao-ora01 bin]$ azacsnap -c test --test oracle --configfile
azacsnap.json
BEGIN : Test process started for 'oracle'
BEGIN : Oracle DB tests
PASSED: Successful connectivity to Oracle DB version 1908000000
END   : Test process complete for 'oracle'
[azacsnap@acao-ora01 bin]$
```

5. Run your first snapshot backup.

```
azacsnap -c backup --volume data --prefix ora_test --retention=1
```

Next: Database protection.

## Protect your Oracle database in Azure cloud

Previous: Deployment procedures.

Author(s): Allen Cao, NetApp Solutions Engineering

### Backup Oracle database with snapshot using AzAcSnap tool

The Azure Application-Consistent Snapshot tool (AzAcSnap) is a command-line tool that enables data protection for third-party databases by handling all the orchestration required to put them into an application-consistent state before taking a storage snapshot, after which it returns the databases to an operational state.

In the case of Oracle, you put the database in backup mode to take a snapshot and then take the database out of backup mode.

#### Backup data and log volumes

The backup can be set up on the database server host with simple shell script that executes the snapshot command. Then, the script can be scheduled to run from crontab.

Generally, the frequency of backup depends on the desired RTO and RPO. Frequent snapshot creation consumes more storage space. There is a trade off between the frequency of backup and space consumption.

Data volumes typically consume more storage space than log volumes. Therefore, you can take snapshots on data volumes every few hours and more frequent snapshots on log volumes every 15 to 30 minutes.

See the following examples of backup scripts and scheduling.

For data volume snapshots:

```
 # /bin/sh
cd /home/azacsnap/bin
. ~/.bash_profile
azacsnap -c backup --volume data --prefix acao-ora01-data --retention 36
azacsnap -c backup --volume other --prefix acao-ora01-log --retention 250
```

For log volume snapshots:

```
 # /bin/sh
cd /home/azacsnap/bin
. ~/.bash_profile
azacsnap -c backup --volume other --prefix acao-ora01-log --retention 250
```

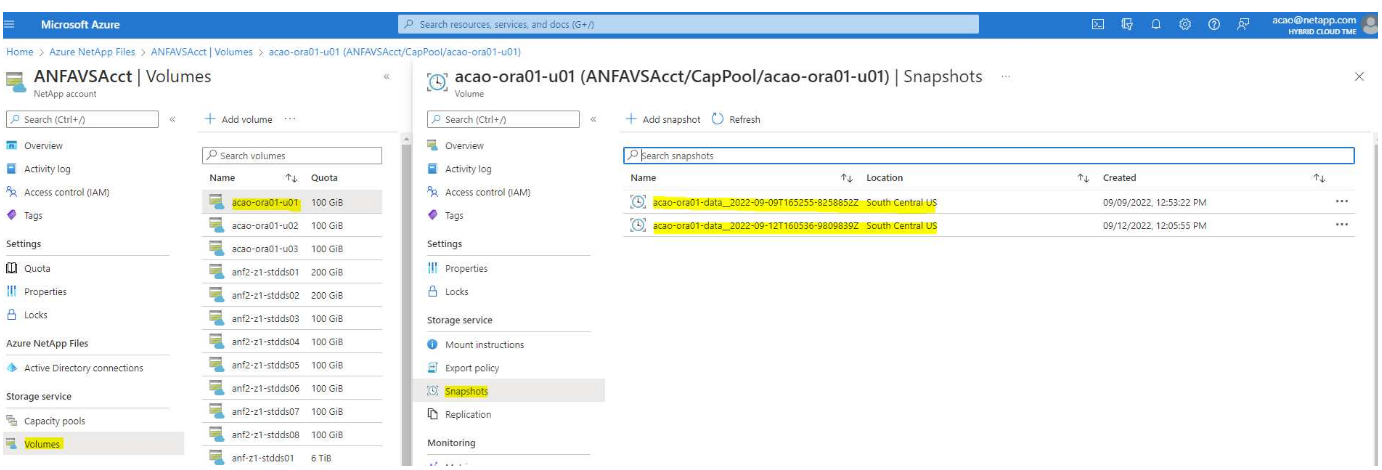
Crontab schedule:

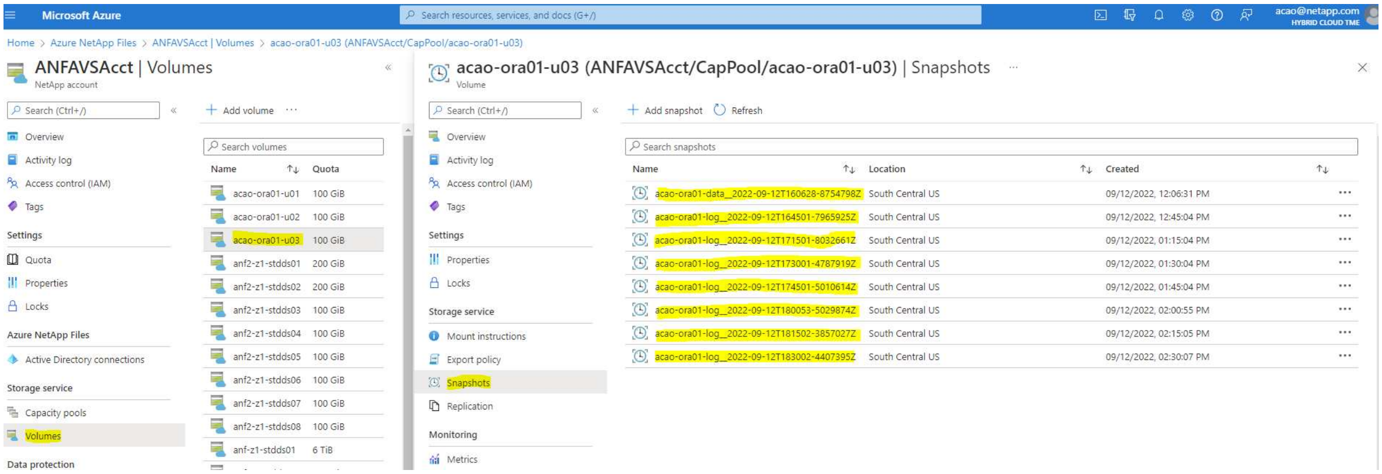
```
15,30,45 * * * * /home/azacsnap/snap_log.sh
0 */2 * * * /home/azacsnap/snap_data.sh
```

> When setting up the backup `azacsnap.json` configuration file, add all data volumes, including the binary volume, to `dataVolume` and all log volumes to `otherVolume`. The maximum retention of snapshots is 250 copies.

#### Validate the snapshots

Go to the Azure portal > Azure NetApp Files/volumes to check if the snapshots have been successfully created.






## Oracle restore and recovery from local backup

One of key benefits of snapshot backup is that it coexists with source database volumes, and the primary database volumes can be rolled back almost instantly.

### Restore and recovery of Oracle on the primary server

The following example demonstrates how to restore and recover an Oracle database from the Azure dashboard and CLI on the same Oracle host.

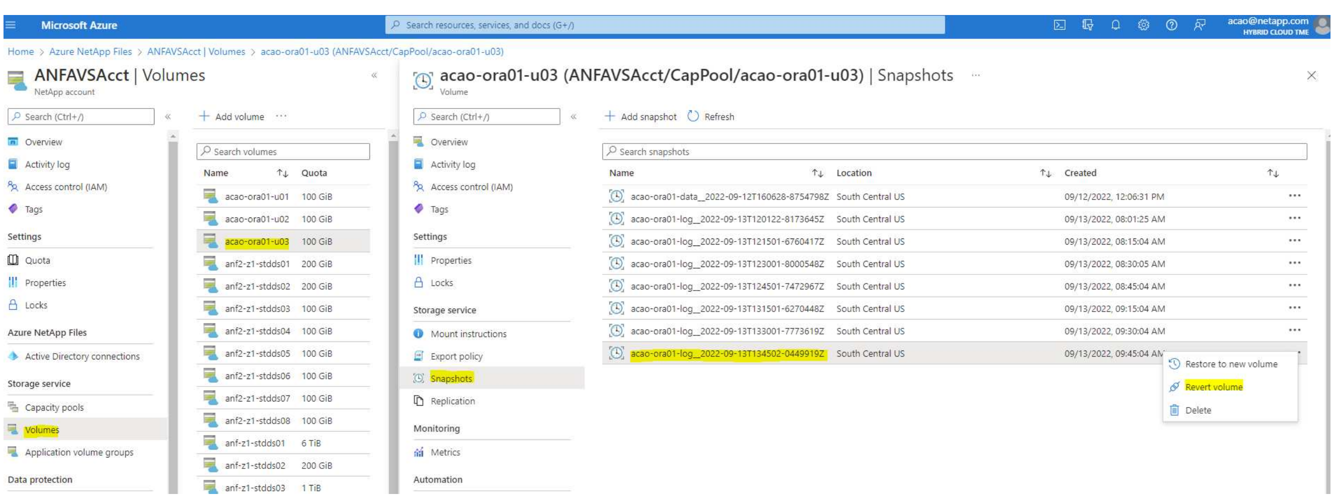
1. Create a test table in the database to be restored.

```
[oracle@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;

 ID
----------
EVENT
------------------------------------------------------------------------
--------
DT
------------------------------------------------------------------------
---
         1
insert a data marker to validate snapshot restore
12-SEP-22 07.07.35.000000 PM
```

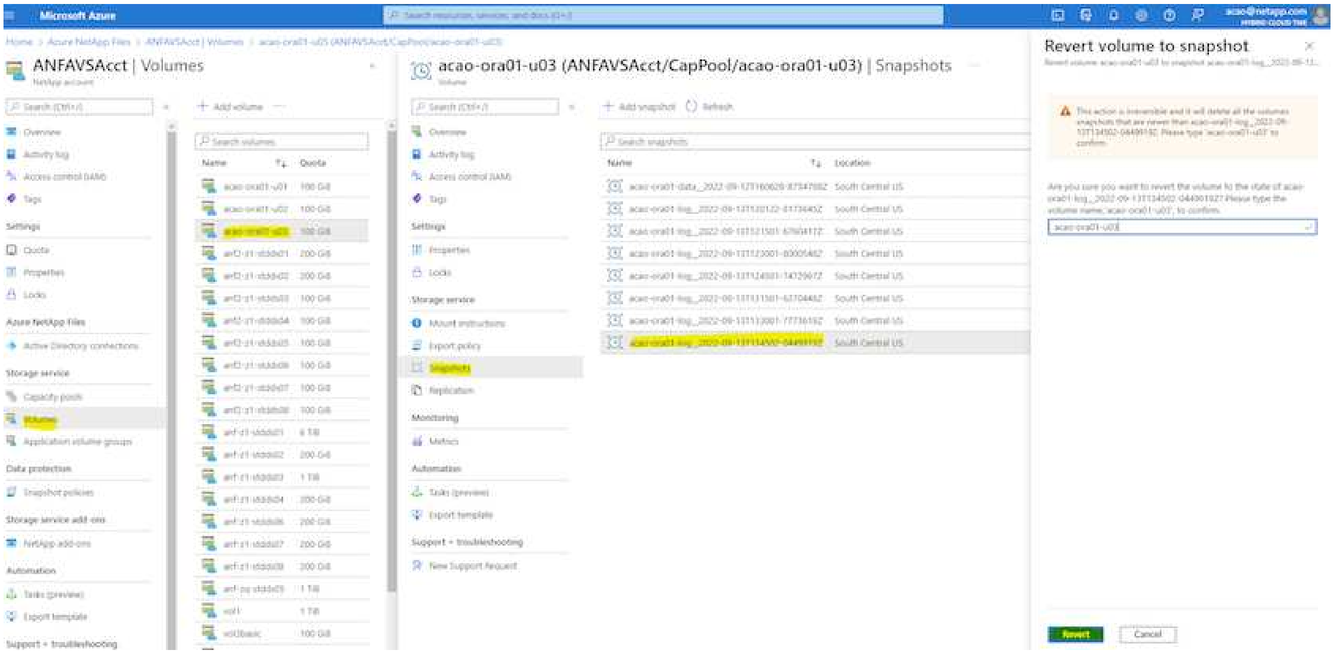
2. Drop the table after the snapshot backups.

```
[oracle@acao-ora01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Tue Sep 13 14:20:22 2022
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL> drop table testsnapshot;

Table dropped.

SQL> select * from testsnapshot;
select * from testsnapshot
              *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.8.0.0.0
```

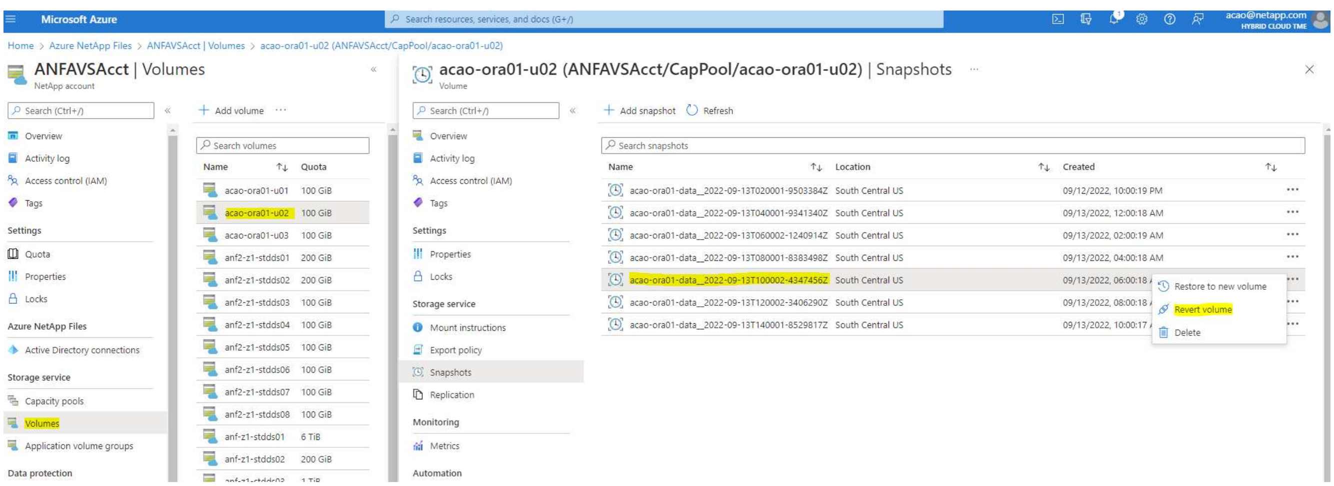
3. From the Azure NetApp Files dashboard, restore the log volume to the last available snapshot. Choose **Revert volume**.



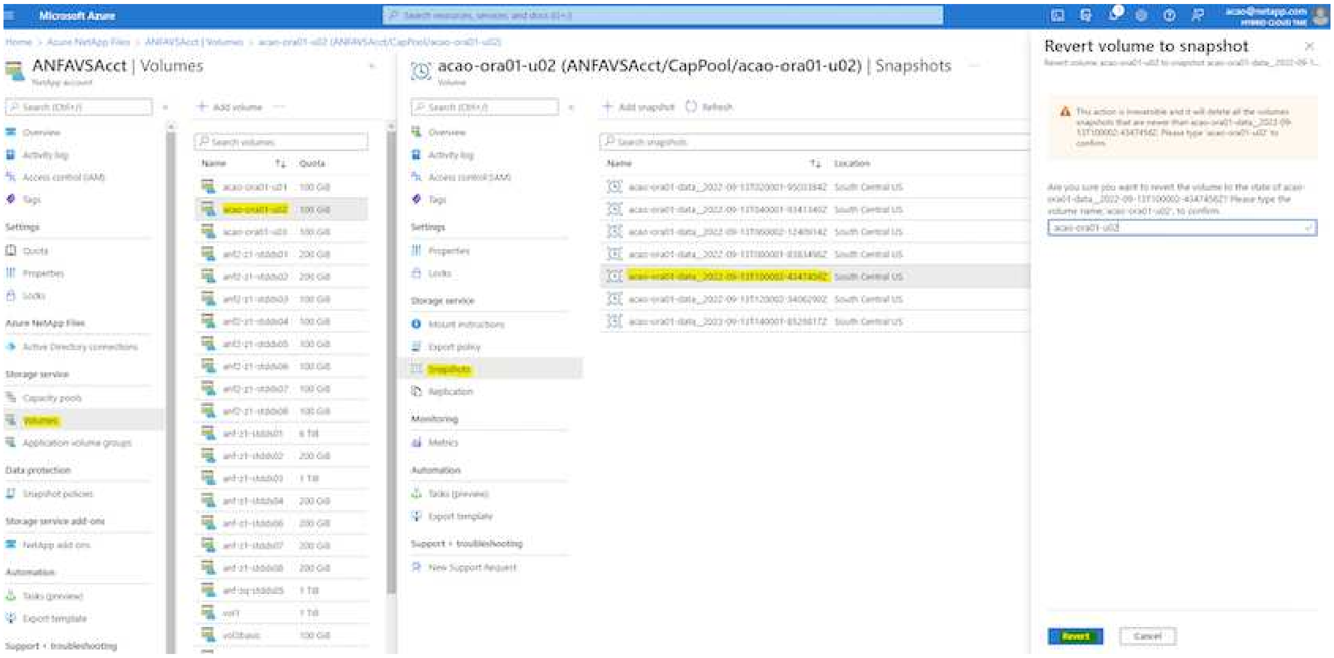
4. Confirm revert volume and click **Revert** to complete the volume reversion to the latest available backup.




5. Repeat the same steps for the data volume, and make sure that the backup contains the table to be recovered.




6. Again confirm the volume reversion, and click "Revert."



7. Resync the control files if you have multiple copies of them, and replace the old control file with the latest copy available.

```
[oracle@acao-ora01 ~]$ mv /u02/oradata/ORATST/control01.ctl
/u02/oradata/ORATST/control01.ctl.bk
[oracle@acao-ora01 ~]$ cp /u03/orareco/ORATST/control02.ctl
/u02/oradata/ORATST/control01.ctl
```

8. Log into the Oracle server VM and run database recovery with sqlplus.

```
[oracle@acao-ora01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Tue Sep 13 15:10:17 2022
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup mount;
ORACLE instance started.

Total System Global Area 6442448984 bytes
Fixed Size                  8910936 bytes
Variable Size            1090519040 bytes
Database Buffers         5335154688 bytes
Redo Buffers                7864320 bytes
Database mounted.
```

```
SQL> recover database using backup controlfile until cancel;
ORA-00279: change 3188523 generated at 09/13/2022 10:00:09 needed for
thread 1
ORA-00289: suggestion :
/u03/orareco/ORATST/archivelog/2022_09_13/o1_mf_1_43__22rnjq9q_.arc
ORA-00280: change 3188523 for thread 1 is in sequence #43

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 3188862 generated at 09/13/2022 10:01:20 needed for
thread 1
ORA-00289: suggestion :
/u03/orareco/ORATST/archivelog/2022_09_13/o1_mf_1_44__29f2lgb5_.arc
ORA-00280: change 3188862 for thread 1 is in sequence #44
ORA-00278: log file
'/u03/orareco/ORATST/archivelog/2022_09_13/o1_mf_1_43__22rnjq9q_.arc' no
longer
needed for this recovery

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 3193117 generated at 09/13/2022 12:00:08 needed for
thread 1
ORA-00289: suggestion :
/u03/orareco/ORATST/archivelog/2022_09_13/o1_mf_1_45__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_44__29f2lgb5_.arc' no
longer
needed for this recovery

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 3193440 generated at 09/13/2022 12:01:20 needed for
thread 1
ORA-00289: suggestion :
/u03/orareco/ORATST/archivelog/2022_09_13/o1_mf_1_46_%u_.arc
ORA-00280: change 3193440 for thread 1 is in sequence #46
ORA-00278: log file
'/u03/orareco/ORATST/archivelog/2022_09_13/o1_mf_1_45__29h6qqyw_.arc' no
longer
needed for this recovery

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
cancel
Media recovery cancelled.
```

```
SQL> alter database open resetlogs;

Database altered.

SQL> select * from testsnapshot;

  ID
----------
EVENT
--------------------------------------------------------------------------
--------
DT
--------------------------------------------------------------------------
---
         1
insert a data marker to validate snapshot restore
12-SEP-22 07.07.35.000000 PM

SQL> select systimestamp from dual;

 SYSTIMESTAMP
--------------------------------------------------------------------------
---
13-SEP-22 03.28.52.646977 PM +00:00
```

This screen demonstrates that the dropped table has been recovered using local snapshot backups.

Next: Database migration.

## Database migration from on-premises to Azure cloud

Previous: Database protection.

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 or PDB.

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.

```
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.

```
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 uses PDB hot-clone technology, which enables source PDB availability while the PDB is being copied over to the target. Upon switchover, sessions and connections are redirected to the target PDB automatically. Thus, down time is minimized independent of the size of the PDB being relocated. 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.

```
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.

```
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.

```
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.

```
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

## TR-3633: Oracle databases on ONTAP

Jeffrey Steiner, NetApp

Consult the Interoperability Matrix Tool (IMT) to determine whether the environment, configurations, and versions specified in TR-3633 support your environment.

TR-3633: Oracle databases on ONTAP

## 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.

#### AWX/Tower Deployments

Part 1: Getting Started, Requirements, Automation Details and Initial AWX/Tower Configuration

AWX Deployment

Part 2: Variables and Running the Playbook

AWX Playbook Run

#### CLI Deployment

Part 1: Getting Started, Requirements, Automation Details and Ansible Control Host Setup

CLI Deployment

Part 2: Variables and Running the Playbook

CLI Playbook Run

### 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

| Environment | Requirements |
|---|---|
| **Ansible environment** | AWX/Tower or Linux host to be the Ansible control host |
| | Ansible v.2.10 and higher |
| | Python 3 |
| | Python libraries<br>- netapp-lib<br>- xmltodict<br>- jmespath |
| **ONTAP** | ONTAP version 9.3 - 9.7 |
| | Two data aggregates |
| | NFS vlan and ifgrp created |
| **Oracle server(s)** | RHEL 7/8 |
| | Oracle Linux 7/8 |
| | Network interfaces for NFS, public, and optional mgmt |
| | Oracle installation files on Oracle servers |

#### 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.

| Role | Tasks |
|---|---|
| **ontap_config** | Pre-check of the ONTAP environment |
| | Creation of NFS based SVM for Oracle |
| | Creation of export policy |
| | Creation of volumes for Oracle |
| | Creation of NFS LIFs |

| Role | Tasks |
|---|---|
| **linux_config** | Create mount points and mount NFS volumes |
| | Verify NFS mounts |
| | OS specific configuration |
| | Create Oracle directories |
| | Configure hugepages |
| | Disable SELinux and firewall daemon |
| | Enable and start chronyd service |
| | increase file descriptor hard limit |
| | Create pam.d session file |
| **oracle_config** | Oracle software installation |
| | Create Oracle listener |
| | Create Oracle databases |
| | Oracle environment configuration |
| | Save PDB state |
| | Enable instance archive mode |
| | Enable DNFS client |
| | Enable database auto startup and shutdown between OS reboots |

#### 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

```
######################################################################
##############      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"
pdb_num: "3"
pdb_name: "{{ oracle_sid }}_pdb"

# CDB listener port, use different listener port for additional CDB on
same host
listener_port: "1523"

# CDB is created with SGA at 75% of memory_limit, MB. Consider how many
databases to be hosted on the node and how much ram to be allocated to
each DB. The grand total SGA should not exceed 75% available RAM on node.
memory_limit: "5464"

# Set "em_configuration: DBEXPRESS" to install enterprise manager express
and choose a unique port from 5500 to 5599 for each sid on the host.
# Leave them black if em express is not installed.
em_configuration: "DBEXPRESS"
em_express_port: "5501"

# {{groups.oracle[0]}} represents first Oracle DB server as defined in
Oracle hosts group [oracle]. For concurrent multiple Oracle DB servers
deployment, [0] will be incremented for each additional DB server. For
example,  {{groups.oracle[1]}}" represents DB server 2,
```

```
"{{groups.oracle[2]}}" represents DB server 3 ... As a good practice and
the default, minimum three volumes is allocated to a DB server with
corresponding /u01, /u02, /u03 mount points, which store oracle binary,
oracle data, and oracle recovery files respectively. Additional volumes
can be added by click on "More NFS volumes" but the number of volumes
allocated to a DB server must match with what is defined in global vars
file by volumes_nfs parameter, which dictates how many volumes are to be
created for each DB server.
host_datastores_nfs:
  - {vol_name: "{{groups.oracle[0]}}_u01", aggr_name: "aggr01_node01",
lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u02", aggr_name: "aggr01_node01",
lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u03", aggr_name: "aggr01_node01",
lif: "172.21.94.200", size: "25"}
```

a. Fill in all variables in the blue fields.

b. After completing variables input, click the Copy button on the form to copy all variables to be transferred to AWX or Tower.

c. Navigate back to AWX or Tower and go to Resources → Hosts, and select and open the Oracle server configuration page.

d. Under the Details tab, click edit and paste the copied variables from step 1 to the Variables field under the YAML tab.

e. Click Save.

f. Repeat this process for any additional Oracle servers in the system.

#### 4. Configure global variables

Variables defined in this section apply to all Oracle hosts, databases, and the ONTAP cluster.

1. Input your environment-specific parameters in following embedded global variables or vars form.

> The items in blue must be changed to match your environment.

```
#######################################################################
###### 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"}
More Storage VLANsEnter Storage VLANs details

#Details of the Data Aggregates that need to be created
#If Aggregate creation takes longer, subsequent tasks of creating volumes
may fail.
#There should be enough disks already zeroed in the cluster, otherwise
aggregate create will zero the disks and will take long time
data_aggregates:
  - {aggr_name: "aggr01_node01"}
  - {aggr_name: "aggr01_node02"}

#SVM name
svm_name: "ora_svm"

# SVM Management LIF Details
svm_mgmt_details:
  - {address: "172.21.91.100", netmask: "255.255.255.0", home_port: "e0M"}

# NFS storage parameters when data_protocol set to NFS. Volume named after
Oracle hosts name identified by mount point as follow for oracle DB server
1. Each mount point dedicates to a particular Oracle files: u01 - Oracle
binary, u02 - Oracle data, u03 - Oracle redo. Add additional volumes by
click on "More NFS volumes" and also add the volumes list to corresponding
host_vars as host_datastores_nfs variable. For multiple DB server
deployment, additional volumes sets needs to be added for additional DB
server. Input variable "{{groups.oracle[1]}}_u01",
"{{groups.oracle[1]}}_u02", and "{{groups.oracle[1]}}_u03" as vol_name for
second DB server. Place volumes for multiple DB servers alternatingly
between controllers for balanced IO performance, e.g. DB server 1 on
```

```
controller node1, DB server 2 on controller node2 etc. Make sure match lif
address with controller node.

volumes_nfs:
  - {vol_name: "{{groups.oracle[0]}}_u01", aggr_name: "aggr01_node01",
lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u02", aggr_name: "aggr01_node01",
lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u03", aggr_name: "aggr01_node01",
lif: "172.21.94.200", size: "25"}

#NFS LIFs IP address and netmask

nfs_lifs_details:
  - address: "172.21.94.200" #for node-1
    netmask: "255.255.255.0"
  - address: "172.21.94.201" #for node-2
    netmask: "255.255.255.0"

#NFS client match

client_match: "172.21.94.0/24"

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

#NFS Mount points for Oracle DB volumes

mount_points:
  - "/u01"
  - "/u02"
  - "/u03"

# Up to 75% of node memory size divided by 2mb. Consider how many
databases to be hosted on the node and how much ram to be allocated to
each DB.
# Leave it blank if hugepage is not configured on the host.

hugepages_nr: "1234"

# RedHat subscription username and password

redhat_sub_username: "xxx"
redhat_sub_password: "xxx"

####################################################
```

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

db_domain: "your.domain.com"

# Set initial password for all required Oracle passwords. Change them
after installation.

initial_pwd_all: "netapp123"
```

1. Fill in all variables in blue fields.
2. After completing variables input, click the Copy button on the form to copy all variables to be transferred to AWX or Tower into the following job template.

**5. Configure and launch the job template.**

1. Create the job template.
   a. Navigate to Resources → Templates → Add and click Add Job Template.
   b. Enter the name and description
   c. Select the Job type; Run configures the system based on a playbook, and Check performs a dry run of a playbook without actually configuring the system.
   d. Select the corresponding inventory, project, playbook, and credentials for the playbook.
   e. Select the all_playbook.yml as the default playbook to be executed.
   f. Paste global variables copied from step 4 into the Template Variables field under the YAML tab.
   g. Check the box Prompt on Launch in the Job Tags field.
   h. Click Save.
2. Launch the job template.
   a. Navigate to Resources → Templates.
   b. Click the desired template and then click Launch.
   c. When prompted on launch for Job Tags, type in requirements_config. You might need to click the Create Job Tag line below requirements_config to enter the job tag.

> ⓘ requirements_config ensures that you have the correct libraries to run the other roles.

a. Click Next and then Launch to start the job.
b. Click View → Jobs to monitor the job output and progress.
c. When prompted on launch for Job Tags, type in ontap_config. You might need to click the Create "Job Tag" line right below ontap_config to enter the job tag.
d. Click Next and then Launch to start the job.
e. Click View → Jobs to monitor the job output and progress
f. After the ontap_config role has completed, run the process again for linux_config.
g. Navigate to Resources → Templates.

h. Select the desired template and then click Launch.
i. When prompted on launch for the Job Tags type in linux_config, you might need to select the Create "job tag" line right below linux_config to enter the job tag.
j. Click Next and then Launch to start the job.
k. Select View → Jobs to monitor the job output and progress.
l. After the linux_config role has completed, run the process again for oracle_config.
m. Go to Resources → Templates.
n. Select the desired template and then click Launch.
o. When prompted on launch for Job Tags, type oracle_config. You might need to select the Create "Job Tag" line right below oracle_config to enter the job tag.
p. Click Next and then Launch to start the job.
q. Select View → Jobs to monitor the job output and progress.

### 6. Deploy additional database on same Oracle host

The Oracle portion of the playbook creates a single Oracle container database on an Oracle server per execution. To create additional container databases on the same server, complete the following steps.

1. Revise host_vars variables.
    a. Go back to step 2 - Configure Oracle host_vars.
    b. Change the Oracle SID to a different naming string.
    c. Change the listener port to different number.
    d. Change the EM Express port to a different number if you are installing EM Express.
    e. Copy and paste the revised host variables to the Oracle Host Variables field in the Host Configuration Detail tab.
2. Launch the deployment job template with only the oracle_config tag.
3. Log in to Oracle server as oracle user and execute the following commands:

```
ps -ef | grep ora
```

> This will list oracle processes if installation completed as expected and oracle DB started

4. Log in to the database to check the db configuration settings and the PDBs created with the following command sets.

```
[oracle@localhost ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Thu May 6 12:52:51 2021
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL>

SQL> select name, log_mode from v$database;
NAME      LOG_MODE
--------- ------------
CDB2      ARCHIVELOG

SQL> show pdbs

    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 CDB2_PDB1                      READ WRITE NO
         4 CDB2_PDB2                      READ WRITE NO
         5 CDB2_PDB3                      READ WRITE NO

col svrname form a30
col dirname form a30
select svrname, dirname, nfsversion from v$dnfs_servers;

SQL> col svrname form a30
SQL> col dirname form a30
SQL> 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 hte 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:

```
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:

```
#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

```
######################################################################
##############      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"
pdb_num: "3"
pdb_name: "{{ oracle_sid }}_pdb"

# CDB listener port, use different listener port for additional CDB on
same host
listener_port: "1523"

# CDB is created with SGA at 75% of memory_limit, MB. Consider how many
databases to be hosted on the node and how much ram to be allocated to
each DB. The grand total SGA should not exceed 75% available RAM on node.
memory_limit: "5464"

# Set "em_configuration: DBEXPRESS" to install enterprise manager express
and choose a unique port from 5500 to 5599 for each sid on the host.
# Leave them black if em express is not installed.
em_configuration: "DBEXPRESS"
em_express_port: "5501"

# {{groups.oracle[0]}} represents first Oracle DB server as defined in
Oracle hosts group [oracle]. For concurrent multiple Oracle DB servers
deployment, [0] will be incremented for each additional DB server. For
example,  {{groups.oracle[1]}}" represents DB server 2,
"{{groups.oracle[2]}}" represents DB server 3 ... As a good practice and
the default, minimum three volumes is allocated to a DB server with
corresponding /u01, /u02, /u03 mount points, which store oracle binary,
oracle data, and oracle recovery files respectively. Additional volumes
can be added by click on "More NFS volumes" but the number of volumes
allocated to a DB server must match with what is defined in global vars
file by volumes_nfs parameter, which dictates how many volumes are to be
created for each DB server.
host_datastores_nfs:
  - {vol_name: "{{groups.oracle[0]}}_u01", aggr_name: "aggr01_node01",
lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u02", aggr_name: "aggr01_node01",
lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u03", aggr_name: "aggr01_node01",
lif: "172.21.94.200", size: "25"}
```

### Edit the vars.yml file

The `vars.yml` file consolidates all environment-specific variables (ONTAP, Linux, or Oracle) for Oracle deployment.

1. Edit and copy the variables from the VARS section and paste these variables into your `vars.yml` file.

```
#######################################################################
###### 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"}
More Storage VLANsEnter Storage VLANs details

#Details of the Data Aggregates that need to be created
#If Aggregate creation takes longer, subsequent tasks of creating volumes
may fail.
#There should be enough disks already zeroed in the cluster, otherwise
aggregate create will zero the disks and will take long time
data_aggregates:
  - {aggr_name: "aggr01_node01"}
  - {aggr_name: "aggr01_node02"}

#SVM name
```

```
svm_name: "ora_svm"

# SVM Management LIF Details
svm_mgmt_details:
  - {address: "172.21.91.100", netmask: "255.255.255.0", home_port: "e0M"}

# NFS storage parameters when data_protocol set to NFS. Volume named after
Oracle hosts name identified by mount point as follow for oracle DB server
1. Each mount point dedicates to a particular Oracle files: u01 - Oracle
binary, u02 - Oracle data, u03 - Oracle redo. Add additional volumes by
click on "More NFS volumes" and also add the volumes list to corresponding
host_vars as host_datastores_nfs variable. For multiple DB server
deployment, additional volumes sets needs to be added for additional DB
server. Input variable "{{groups.oracle[1]}}_u01",
"{{groups.oracle[1]}}_u02", and "{{groups.oracle[1]}}_u03" as vol_name for
second DB server. Place volumes for multiple DB servers alternatingly
between controllers for balanced IO performance, e.g. DB server 1 on
controller node1, DB server 2 on controller node2 etc. Make sure match lif
address with controller node.

volumes_nfs:
  - {vol_name: "{{groups.oracle[0]}}_u01", aggr_name: "aggr01_node01",
lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u02", aggr_name: "aggr01_node01",
lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u03", aggr_name: "aggr01_node01",
lif: "172.21.94.200", size: "25"}

#NFS LIFs IP address and netmask

nfs_lifs_details:
  - address: "172.21.94.200" #for node-1
    netmask: "255.255.255.0"
  - address: "172.21.94.201" #for node-2
    netmask: "255.255.255.0"

#NFS client match

client_match: "172.21.94.0/24"

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

#NFS Mount points for Oracle DB volumes

mount_points:
```

```
  - "/u01"
  - "/u02"
  - "/u03"

# Up to 75% of node memory size divided by 2mb. Consider how many
databases to be hosted on the node and how much ram to be allocated to
each DB.
# Leave it blank if hugepage is not configured on the host.

hugepages_nr: "1234"

# RedHat subscription username and password

redhat_sub_username: "xxx"
redhat_sub_password: "xxx"

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

db_domain: "your.domain.com"

# Set initial password for all required Oracle passwords. Change them
after installation.

initial_pwd_all: "netapp123"
```

#### 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.

<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;

SQL> col svrname form a30
SQL> col dirname form a30
SQL> select svrname, dirname, nfsversion from v$dnfs_servers;

SVRNAME                        DIRNAME                        NFSVERSION
------------------------------ ------------------------------
----------------
172.21.126.200                 /rhelora03_u02                 NFSv3.0
172.21.126.200                 /rhelora03_u03                 NFSv3.0
172.21.126.200                 /rhelora03_u01                 NFSv3.0
```

This confirms that dNFS is working properly.

3. Connect to database via listener to check hte 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.

# Oracle Database Data Protection

## 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

For more details or to begin, please see the overview videos below.

#### AWX/Tower Deployments

- Part 1: TBD

  **video**
- Part 2: TBD

  **video**

  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**

## On-Prem |

| Environment | Requirements |
| --- | --- |
| **Ansible environment** | AWX/Tower |
| | Ansible v.2.10 and higher |
| | Python 3 |
| | Python libraries<br>- netapp-lib<br>- xmltodict<br>- jmespath |
| **ONTAP** | ONTAP version 9.8 + |
| | Two data aggregates |
| | NFS vlan and ifgrp created |
| **Oracle server(s)** | RHEL 7/8 |
| | Oracle Linux 7/8 |
| | Network interfaces for NFS, public, and optional mgmt |
| | Existing Oracle environment on source, and the equivalent Linux operating system at the destination (DR Site or Public Cloud) |

## CVO

| Environment | Requirements |
| --- | --- |
| **Ansible environment** | AWX/Tower |
| | Ansible v.2.10 and higher |
| | Python 3 |
| | Python libraries<br>- netapp-lib<br>- xmltodict<br>- jmespath |
| **ONTAP** | ONTAP version 9.8 + |
| | Two data aggregates |
| | NFS vlan and ifgrp created |
| **Oracle server(s)** | |
| | RHEL 7/8 |
| | Oracle Linux 7/8 |
| | Network interfaces for NFS, public, and optional mgmt |
| | Existing Oracle environment on source, and the equivalent Linux operating system at the destination (DR Site or Public Cloud) |
| | Set appropriate swap space on the Oracle EC2 instance, by default some EC2 instances are deployed with 0 swap |

| Environment | Requirements |
| --- | --- |
| **Cloud Manager/AWS** | AWS Access/Secret Key |
| | NetApp Cloud Manager Account |
| | NetApp Cloud Manager Refresh Token |

#### 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.

| Playbook | Tasks |
|---|---|
| **ontap_setup** | Pre-check of the ONTAP environment |
| | Creation of Intercluster LIFs on source cluster (OPTIONAL) |
| | Creation of Intercluster LIFs on destination cluster (OPTIONAL) |
| | Creation of Cluster and SVM Peering |
| | Creation of destination SnapMirror and Initialization of designated Oracle volumes |
| **ora_replication_cg** | Enable backup mode for each database in /etc/oratab |
| | Snapshot taken of Oracle Binary and Database volumes |
| | Snapmirror Updated |
| | Turn off backup mode for each database in /etc/oratab |
| **ora_replication_log** | Switch current log for each database in /etc/oratab |
| | Snapshot taken of Oracle Log volume |
| | Snapmirror Updated |
| **ora_recovery** | Break SnapMirror |
| | Enable NFS and create junction path for Oracle volumes on the destination |
| | Configure DR Oracle Host |
| | Mount and verify Oracle volumes |
| | Recover and start Oracle database |

## 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.

| Playbook | Tasks |
| --- | --- |
| **cvo_setup** | Pre-check of the environment |
| | AWS Configure/AWS Access Key ID/Secret Key/Default Region |
| | Creation of AWS Role |
| | Creation of NetApp Cloud Manager Connector instance in AWS |
| | Creation of Cloud Volumes ONTAP (CVO) instance in AWS |
| | Add On-Prem Source ONTAP Cluster to NetApp Cloud Manager |
| | Creation of destination SnapMirror and Initialization of designated Oracle volumes |
| **ora_replication_cg** | Enable backup mode for each database in /etc/oratab |
| | Snapshot taken of Oracle Binary and Database volumes |
| | Snapmirror Updated |
| | Turn off backup mode for each database in /etc/oratab |
| **ora_replication_log** | Switch current log for each database in /etc/oratab |
| | Snapshot taken of Oracle Log volume |
| | Snapmirror Updated |
| **ora_recovery** | Break SnapMirror |
| | Enable NFS and create junction path for Oracle volumes on the destination CVO |
| | Configure DR Oracle Host |
| | Mount and verify Oracle volumes |
| | Recover and start Oracle database |

#### 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.

**On-Prem**

```
# 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**

```
###########################################
### Ontap env specific config variables ###
###########################################

#Inventory group name
#Default inventory group name - "ontap"
#Change only if you are changing the group name either in
inventory/hosts file or in inventory groups in case of AWX/Tower
hosts_group: "ontap"

#CA_signed_certificates (ONLY CHANGE to "true" IF YOU ARE USING CA
SIGNED CERTIFICATES)
ca_signed_certs: "false"

#Names of the Nodes in the Source ONTAP Cluster
src_nodes:
  - "AFF-01"
  - "AFF-02"

#Names of the Nodes in the Destination CVO Cluster
```

```
dst_nodes:
  - "DR-AFF-01"
  - "DR-AFF-02"

#Define whether or not to create intercluster lifs on source cluster
(ONLY CHANGE to "No" IF YOU HAVE ALREADY CREATED THE INTERCLUSTER LIFS)
create_source_intercluster_lifs: "yes"

source_intercluster_network_port_details:
  using_dedicated_ports: "yes"
  using_ifgrp: "yes"
  using_vlans: "yes"
  failover_for_shared_individual_ports: "yes"
  ifgrp_name: "a0a"
  vlan_id: "10"
  ports:
    - "e0b"
    - "e0g"
  broadcast_domain: "NFS"
  ipspace: "Default"
  failover_group_name: "iclifs"

source_intercluster_lif_details:
  - name: "icl_1"
    address: "10.0.0.1"
    netmask: "255.255.255.0"
    home_port: "a0a-10"
    node: "AFF-01"
  - name: "icl_2"
    address: "10.0.0.2"
    netmask: "255.255.255.0"
    home_port: "a0a-10"
    node: "AFF-02"

###########################################
### CVO Deployment Variables ###
###########################################

####### Access Keys Variables ######

# Region where your CVO will be deployed.
region_deploy: "us-east-1"

########### CVO and Connector Vars ########

# AWS Managed Policy required to give permission for IAM role creation.
```

```
aws_policy: "arn:aws:iam::1234567:policy/OCCM"

# Specify your aws role name, a new role is created if one already does
not exist.
aws_role_name: "arn:aws:iam::1234567:policy/OCCM"

# Name your connector.
connector_name: "awx_connector"

# Name of the key pair generated in AWS.
key_pair: "key_pair"

# Name of the Subnet that has the range of IP addresses in your VPC.
subnet: "subnet-12345"

# ID of your AWS secuirty group that allows access to on-prem
resources.
security_group: "sg-123123123"

# You Cloud Manager Account ID.
account: "account-A23123A"

# Name of the your CVO instance
cvo_name: "test_cvo"

# ID of the VPC in AWS.
vpc: "vpc-123123123"

######################################################################
############################
# Variables for - Add on-prem ONTAP to Connector in Cloud Manager
######################################################################
############################

# For Federated users, Client ID from API Authentication Section of
Cloud Central to generate access token.
sso_id: "123123123123123123123"

# For regular access with username and password, please specify "pass"
as the connector_access. For SSO users, use "refresh_token" as the
variable.
connector_access: "pass"

######################################################################
#############################
# Variables for SnapMirror Peering
######################################################################
```

```
##############################
passphrase: "your-passphrase"

#######################################################################
##############################
# Source & Destination List
#######################################################################
##############################
#Please Enter Destination Cluster Name
dst_cluster_name: "dst-cluster-name"

#Please Enter Destination Cluster (Once CVO is Created Add this
Variable to all templates)
dst_cluster_ip: "dst-cluster-ip"

#Please Enter Destination SVM to create mirror relationship
dst_vserver: "dst-vserver"

#Please Enter NFS Lif for dst vserver (Once CVO is Created Add this
Variable to all templates)
dst_nfs_lif: "dst-nfs-lif"

#Please Enter Source Cluster Name
src_cluster_name: "src-cluster-name"

#Please Enter Source Cluster
src_cluster_ip: "src-cluster-ip"

#Please Enter Source SVM
src_vserver: "src-vserver"

#######################################################################
##############################
# Variable for Oracle Volumes and SnapMirror Details
#######################################################################
##############################
#Please Enter Source Snapshot Prefix Name
cg_snapshot_name_prefix: "oracle"

#Please Enter Source Oracle Binary Volume(s)
src_orabinary_vols:
  - "binary_vol"
#Please Enter Source Database Volume(s)
src_db_vols:
  - "db_vol"
#Please Enter Source Archive Volume(s)
```

```
src_archivelog_vols:
  - "log_vol"
#Please Enter Destination Snapmirror Policy
snapmirror_policy: "async_policy_oracle"

#######################################################################
##############################
# Export Policy Details
#######################################################################
##############################
#Enter the destination export policy details (Once CVO is Created Add
this Variable to all templates)
export_policy_details:
  name: "nfs_export_policy"
  client_match: "0.0.0.0/0"
  ro_rule: "sys"
  rw_rule: "sys"

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

#NFS Mount points for Oracle DB volumes
mount_points:
  - "/u01"
  - "/u02"
  - "/u03"

# Up to 75% of node memory size divided by 2mb. Consider how many
databases to be hosted on the node and how much ram to be allocated to
each DB.
# Leave it blank if hugepage is not configured on the host.
hugepages_nr: "1234"

# RedHat subscription username and password
redhat_sub_username: "xxx"
redhat_sub_password: "xxx"

####################################################
### DB env specific install and config variables ###
####################################################
#Recovery Type (leave as scn)
recovery_type: "scn"
```

```
#Oracle Control Files
control_files:
  - "/u02/oradata/CDB2/control01.ctl"
  - "/u03/orareco/CDB2/control02.ctl"
```

#### Automation Playbooks

There are four separate playbooks that need to be ran.

1. Playbook for Setting up your environment, On-Prem or CVO.
2. Playbook for replicating Oracle Binaries and Databases on a schedule
3. Playbook for replicating Oracle Logs on a schedule
4. Playbook for Recovering your database on a destination host

**ONTAP/CVO Setup**

ONTAP and CVO Setup

**Configure and launch the job template.**

1. Create the job template.
    a. Navigate to Resources → Templates → Add and click Add Job Template.
    b. Enter the name ONTAP/CVO Setup
    c. Select the Job type; Run configures the system based on a playbook.
    d. Select the corresponding inventory, project, playbook, and credentials for the playbook.
    e. Select the ontap_setup.yml playbook for an On-Prem environment or select the cvo_setup.yml for replicating to a CVO instance.
    f. Paste global variables copied from step 4 into the Template Variables field under the YAML tab.
    g. Click Save.
2. Launch the job template.
    a. Navigate to Resources → Templates.
    b. Click the desired template and then click Launch.

> We will use this template and copy it out for the other playbooks.

**Replication For Binary and Database Volumes**

Scheduling the Binary and Database Replication Playbook

**Configure and launch the job template.**

1. Copy the previously created job template.
    a. Navigate to Resources → Templates.
    b. Find the ONTAP/CVO Setup Template, and on the far right click on Copy Template
    c. Click Edit Template on the copied template, and change the name to Binary and Database Replication Playbook.
    d. Keep the same inventory, project, credentials for the template.
    e. Select the ora_replication_cg.yml as the playbook to be executed.
    f. The variables will remain the same, but the CVO cluster IP will need to be set in the variable dst_cluster_ip.
    g. Click Save.
2. Schedule the job template.
    a. Navigate to Resources → Templates.
    b. Click the Binary and Database Replication Playbook template and then click Schedules at the top set of options.
    c. Click Add, add Name Schedule for Binary and Database Replication, choose the Start date/time at the beginning of the hour, choose your Local time zone, and Run frequency. Run frequency will be often the SnapMirror replication will be updated.

> A separate schedule will be created for the Log volume replication, so that it can be replicated on a more frequent cadence.

**Replication for Log Volumes**

Scheduling the Log Replication Playbook

**Configure and launch the job template.**

1. Copy the previously created job template.
   a. Navigate to Resources → Templates.
   b. Find the ONTAP/CVO Setup Template, and on the far right click on Copy Template
   c. Click Edit Template on the copied template, and change the name to Log Replication Playbook.
   d. Keep the same inventory, project, credentials for the template.
   e. Select the ora_replication_logs.yml as the playbook to be executed.
   f. The variables will remain the same, but the CVO cluster IP will need to be set in the variable dst_cluster_ip.
   g. Click Save.
2. Schedule the job template.
   a. Navigate to Resources → Templates.
   b. Click the Log Replication Playbook template and then click Schedules at the top set of options.
   c. Click Add, add Name Schedule for Log Replication, choose the Start date/time at the beginning of the hour, choose your Local time zone, and Run frequency. Run frequency will be often the SnapMirror replication will be updated.

   > It is recommended to set the log schedule to update every hour to ensure the recovery to the last hourly update.

**Restore and Recover Database**

Scheduling the Log Replication Playbook

**Configure and launch the job template.**

1. Copy the previously created job template.
   a. Navigate to Resources → Templates.
   b. Find the ONTAP/CVO Setup Template, and on the far right click on Copy Template
   c. Click Edit Template on the copied template, and change the name to Restore and Recovery Playbook.
   d. Keep the same inventory, project, credentials for the template.
   e. Select the ora_recovery.yml as the playbook to be executed.
   f. The variables will remain the same, but the CVO cluster IP will need to be set in the variable dst_cluster_ip.
   g. Click Save.

This playbook will not be ran until you are ready to restore your database at the remote site.

**Recovering Oracle Database**

1. On-premises production Oracle databases data volumes are protected via NetApp SnapMirror replication to either a redundant ONTAP cluster in secondary data center or Cloud Volume ONTAP in public cloud. In a fully configured disaster recovery environment, recovery compute instances in secondary data center or public cloud are standby and ready to recover the production database in the case of a disaster. The standby compute instances are kept in sync with on-prem instances by running paraellel 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 sofware 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