



Oracle Database

NetApp Solutions

NetApp
June 05, 2023

This PDF was generated from https://docs.netapp.com/us-en/netapp-solutions/databases/aws_ora_fsx_ec2_nfs_asm.html on June 05, 2023. Always check docs.netapp.com for the latest.

Table of Contents

Oracle Database	1
TR-4974: Oracle 19c in Standalone Restart on AWS FSx/EC2 with NFS/ASM	1
TR-4965: Oracle Database Deployment and Protection in AWS FSx/EC2 with iSCSI/ASM	27
Oracle Database Deployment on AWS EC2 and FSx Best Practices	58
Oracle Database Deployment and Protection on Azure NetApp Files	114
NVA-1155: Oracle 19c RAC databases on FlexPod Datacenter with Cisco UCS and NetApp AFF A800 over FC - Design and deployment guide	153
TR-4250: SAP with Oracle on UNIX and NFS with NetApp Clustered Data ONTAP and SnapManager for SAP 3.4	153
Deploying Oracle Database	153
Oracle Database Data Protection	192
TR-4794: Oracle databases on NetApp EF-Series	232

Oracle Database

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

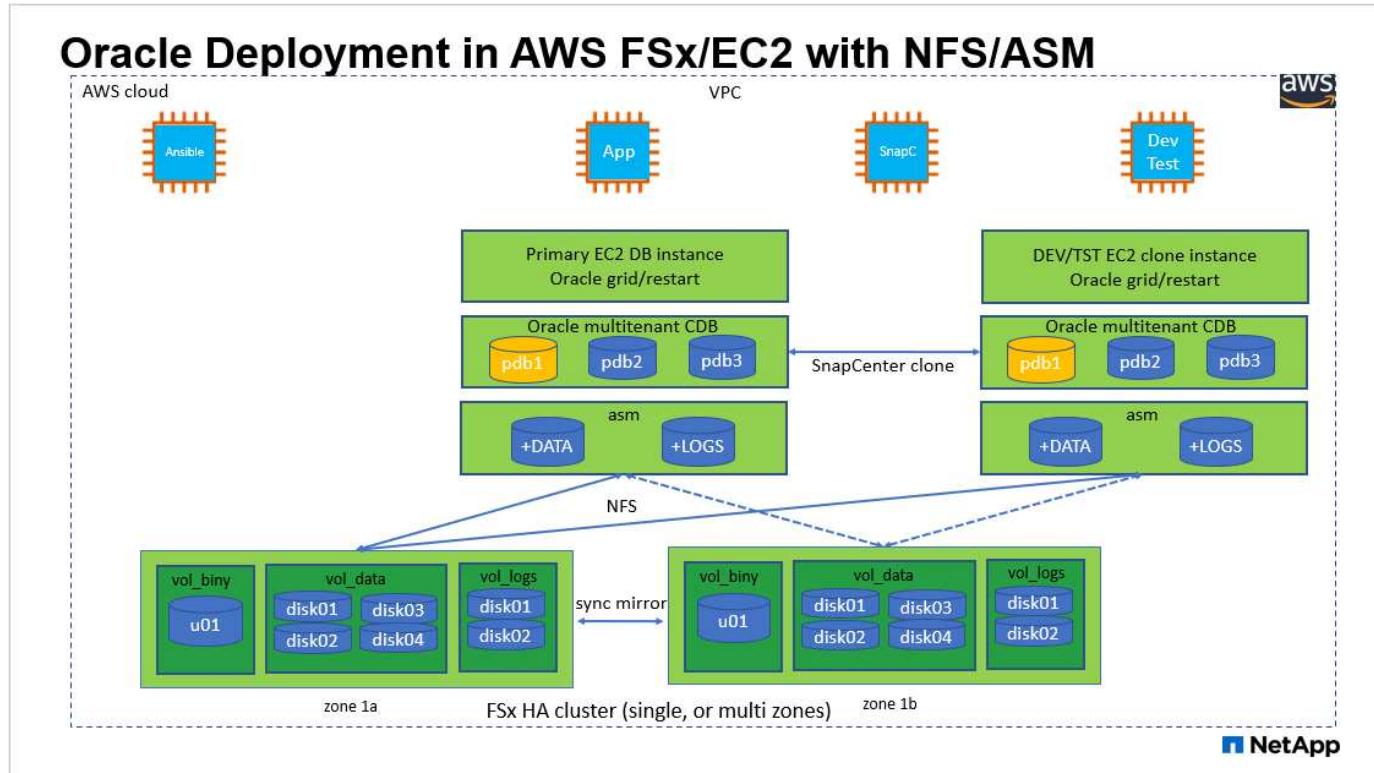
This solution is intended for the following people:

- 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 always provisioned in a HA pair that is sync mirrored in a pair of active-passive file systems to provide storage-level redundancy. Multi-zone deployment further enhances high availability in the event of failure in a single AWS zone.
- **FSx storage cluster sizing.** An Amazon FSx for ONTAP storage file system provides up to 160,000 raw SSD IOPS, up to 4GBps throughput, and a maximum of 192TiB capacity. However, you can size the cluster in terms of provisioned IOPS, throughput, and the storage limit (minimum 1,024 GiB) based on your actual requirements at the time of deployment. The capacity can be adjusted dynamically on the fly without affecting application availability.
- **Oracle data and logs layout.** In our tests and validations, we deployed two ASM disk groups for data and logs respectively. Within the +DATA asm disk group, we provisioned four disks in a data NFS file system mount point. Within the +LOGS asm disk group, we provisioned two disks in a logs NFS file system mount point. For large database deployment, ASM disk groups can be built to span multiple FSx file systems with ASM NFS disks distributed through multiple NFS mount points anchored on FSx file systems. This particular setup is designed to meet database throughput over 4GBps throughput and 160,000 raw SSD IOPS requirement.
- **dNFS configuration.** dNFS is built into Oracle kernel and is known to dramatically increase Oracle database performance when Oracle is deployed to NFS storage. dNFS is packaged into Oracle binary but is not turned on by default. It should be turned on for any Oracle database deployment on NFS. For multiple FSx file systems deployment for large database, dNFS multi-path should be properly configured.
- **Oracle ASM redundancy level to use for each Oracle ASM disk group that you create.** Because FSx already mirrors the storage on the FSx cluster level, you should ONLY use External Redundancy, which means that the option does not allow Oracle ASM to mirror the contents of the disk group. This is particularly important as NFS for Oracle database data storage requires HARD NFS mount option which is NOT desirable for mirroring ASM contents on the Oracle level.
- **Database backup.** NetApp provides a SaaS version of SnapCenter software service for database backup, restore, and clone in the cloud that is available through the NetApp BlueXP console UI. NetApp recommends implementing such a service to achieve fast (under a minute) SnapShot backup, quick (few minutes) database restore, and database cloning.

Solution deployment

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

Prerequisites for deployment

Deployment requires the following prerequisites.

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

```
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_bin -aggregate aggr1 -size 50G -state online -type RW -junction-path /ora_01_bin -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_bin    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_bin /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 /oralog, changed to your FSx NFS lif IP address

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

10. Change /oralog mount point ownership to the Oracle user and its associated primary group.

```
sudo chown oracle:oinstall /oralog
```

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

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

Add the following line.

```
172.30.15.19:/ora_01_bin /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 /oralog 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 /oralog/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=/oradbs/asm/nfs_logs_disk01 bs=1M count=40960
oflag=direct
dd if=/dev/zero of=/oradbs/asm/nfs_logs_disk02 bs=1M count=40960
oflag=direct
```

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

```
chmod 640 /oradbs/asm/*
```

17. Reboot the EC2 instance host.

Oracle grid infrastructure installation

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

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

2. Restart the sshd service.

```
sudo systemctl restart sshd
```

3. Reset the Oracle user password.

```
sudo passwd oracle
```

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

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

5. Change the directory permission setting.

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

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

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

7. Unzip the grid installation files.

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

8. From grid home, delete the OPatch directory.

```
rm -rf OPatch
```

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

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

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

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

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

```
INVENTORY_LOCATION=/u01/app/oraInventory
oracle.install.option=HA_CONFIG
ORACLE_BASE=/u01/app/oracle
oracle.install.asm.OSDBA=dba
oracle.install.asm.OSOPER=oper
oracle.install.asm.OSASM=asm
oracle.install.asm.SYSASMPassword="SetPWD"
oracle.install.asm.diskGroup.name=DATA
oracle.install.asm.diskGroup.redundancy=EXTERNAL
oracle.install.asm.diskGroup.AUSize=4
oracle.install.asm.diskGroup.disks=/oradata/asm/*,/oralog/asm/*
oracle.install.asm.diskGroup.diskDiscoveryString=/oradata/asm/nfs_data_disk01,/oradata/asm/nfs_data_disk02,/oradata/asm/nfs_data_disk03,/oradata/asm/nfs_data_disk04
oracle.install.asm.monitorPassword="SetPWD"
oracle.install.asm.configureAFD=false
```

12. Log into the EC2 instance as the root user.

13. Install `cvuqdisk-1.0.10-1.rpm`.

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

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

```
unzip p34762026_190000_Linux-x86-64.zip
```

15. From grid home `/u01/app/oracle/product/19.0.0/grid` and as the oracle user, launch `gridSetup.sh` for grid infrastructure installation.

```
./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  
'/oratalogs/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 software-only installation.

```
/u01/app/oracle/product/19.0.0/db1/root.sh
```

11. As Oracle user, create the `dbca.rsp` file with the following entries:

```
gdbName=db1.demo.netapp.com
sid=db1
createAsContainerDatabase=true
numberOfPDBs=3
pdbName=db1_pdb
useLocalUndoForPDBs=true
pdbAdminPassword="yourPWD"
templateName=General_Purpose.dbc
sysPassword="yourPWD"
systemPassword="yourPWD"
dbsnmpPassword="yourPWD"
storageType=ASM
diskGroupName=DATA
characterSet=AL32UTF8
nationalCharacterSet=AL16UTF16
listeners=LISTENER
databaseType=MULTIPURPOSE
automaticMemoryManagement=false
totalMemory=8192
```



Set the total memory based on available memory in EC2 instance host. Oracle allocates 75% of `totalMemory` to DB instance SGA or buffer cache.

12. As Oracle user, launch DB creation with dbca.

```
bin/dbca -silent -createDatabase -responseFile /tmp/archive/dbca.rsp

output:
Prepare for db operation
7% complete
Registering database with Oracle Restart
11% complete
Copying database files
33% complete
Creating and starting Oracle instance
35% complete
38% complete
42% complete
45% complete
48% complete
Completing Database Creation
53% complete
55% complete
56% complete
Creating Pluggable Databases
60% complete
64% complete
69% complete
78% complete
Executing Post Configuration Actions
100% complete
Database creation complete. For details check the logfiles at:
/u01/app/oracle/cfgtoollogs/dbca/db1.

Database Information:
Global Database Name:db1.demo.netapp.com
System Identifier(SID):db1
Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/db1/db1.log"
for further details.
```

13. As Oracle user, validate Oracle Restart HA services after DB creation.

```
[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 orafstab 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_bin 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_STATUS 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          /oralog/asm/nfs_logs_disk01 MEMBER
CACHED  NORMAL
LOGS_0001          /oralog/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

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-8225-4784bc9843&sc_channel=ps&s_kwcid=AL!4422!3!467723097970!e!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARIaAjzSrdqwQrghn6I71jiWzSeaT9Uh1-vY-VfhJixFxnv5rWwn2S7RqZOTQ0aAh7eEALw_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

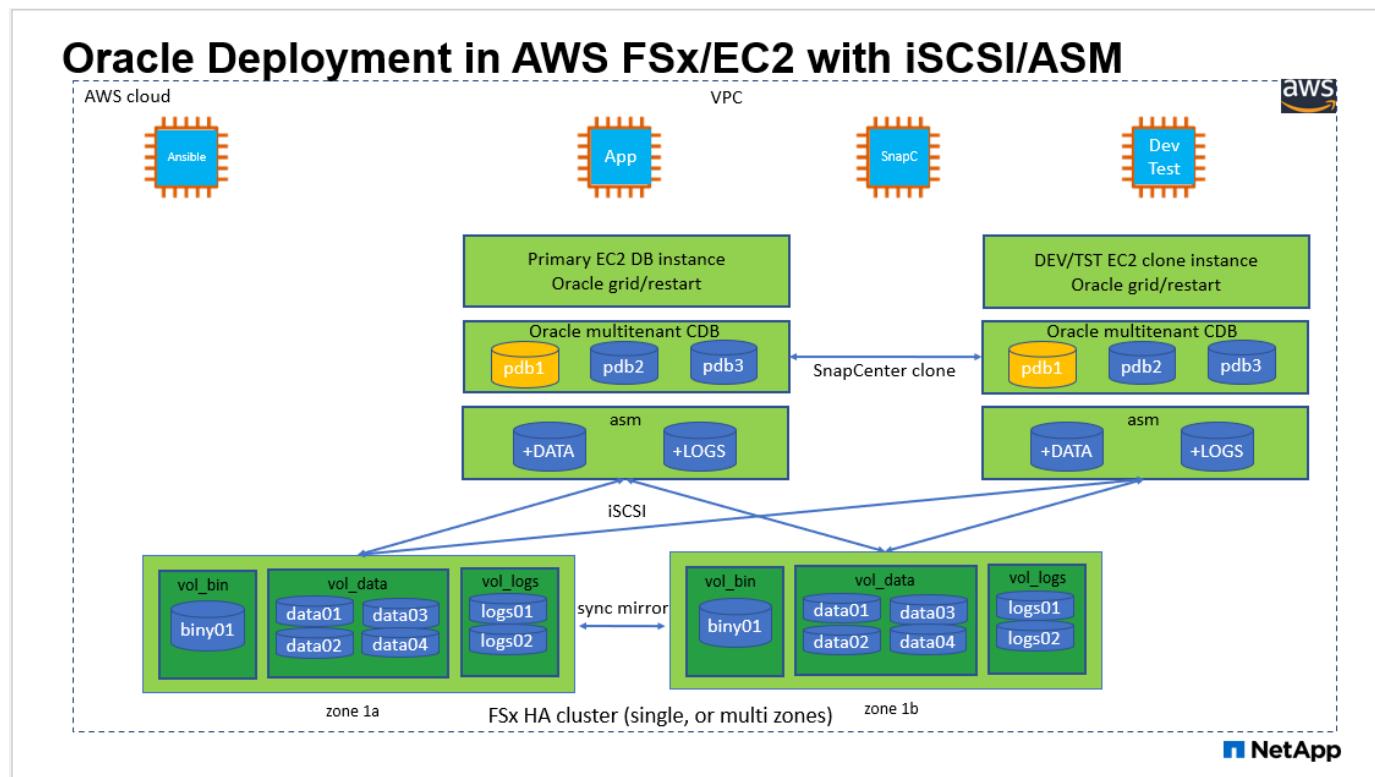
This solution is intended for the following people:

- 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 large 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 always provisioned in a HA pair that is sync mirrored in a pair of active-passive file systems to provide storage-level redundancy. Multi-zone deployment further enhances high availability in the event of failure in a single AWS zone.
- **FSx storage cluster sizing.** An Amazon FSx for ONTAP storage file system provides up to 160,000 raw SSD IOPS, up to 4Gbps throughput, and a maximum of 192TiB capacity. However, you can size the cluster in terms of provisioned IOPS, throughput, and the storage limit (minimum 1,024 GiB) based on your actual requirements at the time of deployment. The capacity can be adjusted dynamically on the fly without affecting application availability.
- **Oracle data and logs layout.** In our tests and validations, we deployed two ASM disk groups for data and logs respectively. Within the +DATA asm disk group, we provisioned four LUNs in a data volume. Within the +LOGS asm disk group, we provisioned two LUNs in a logs volume. In general, multiple LUNs laid out within an Amazon FSx for ONTAP volume provides better performance.
- **iSCSI configuration.** The EC2 instance database server connects to FSx storage with the iSCSI protocol. EC2 instances generally deploy with a single network interface or ENI. The single NIC interface carries both iSCSI and application traffic. It is important to gauge the Oracle database peak I/O throughput requirement by carefully analyzing the Oracle AWR report in order to choose a right EC2 compute instance that meets both application and iSCSI traffic-throughput requirements. NetApp also recommends allocating four iSCSI connections to both FSx iSCSI endpoints with multipath properly configured.
- **Oracle ASM redundancy level to use for each Oracle ASM disk group that you create.** Because FSx already mirrors the storage on the FSx cluster level, you should use External Redundancy, which means that the option does not allow Oracle ASM to mirror the contents of the disk group.
- **Database backup.** NetApp provides a SaaS version of SnapCenter software service for database backup, restore, and clone in the cloud that is available through the NetApp BlueXP console UI. NetApp recommends implementing such a service to achieve fast (under a minute) SnapShot backup, quick (few minutes) database restore, and database cloning.

Solution deployment

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

Prerequisites for deployment

Deployment requires the following prerequisites.

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

```
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.timeout.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_bin -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_bin/ora_01_bin_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.

```
map -path /vol/ora_01_bin/ora_01_bin_01 -igroup ora_01 -vserver  
svm_ora -lun-id 0  
map -path /vol/ora_01_data/ora_01_data_01 -igroup ora_01 -vserver  
svm_ora -lun-id 1  
map -path /vol/ora_01_data/ora_01_data_02 -igroup ora_01 -vserver  
svm_ora -lun-id 2  
map -path /vol/ora_01_data/ora_01_data_03 -igroup ora_01 -vserver  
svm_ora -lun-id 3  
map -path /vol/ora_01_data/ora_01_data_04 -igroup ora_01 -vserver  
svm_ora -lun-id 4  
map -path /vol/ora_01_logs/ora_01_logs_01 -igroup ora_01 -vserver  
svm_ora -lun-id 5  
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_bin/y/ora_01_bin/y_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 iscsiadadm iscsiadadm --mode discovery --op update --type  
sendtargets --portal 172.30.15.51
```

3. Establish iSCSI sessions by logging into each target.

```
sudo iscsiadadm --mode node -l all
```

The expected output from the command is:

```
[ec2-user@ip-172-30-15-58 ~]$ sudo iscsiadadm --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 iscsiadadm --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
}

[source, cli]
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_bin/ora_01_bin_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_bin_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_bin_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_bin_01
```

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

```
sudo mkfs.xfs /dev/mapper/ora_01_bin_01p1
```

13. Mount the binary LUN to /u01.

```
sudo mount -t xfs /dev/mapper/ora_01_bin_01p1 /u01
```

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

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

15. Find the UUID of the binary LUN.

```
sudo blkid /dev/mapper/ora_01_bin_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, copy `p6880880_190000_Linux-x86-64.zip` to the `grid_home`, and then unzip it.

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

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

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

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

```
INVENTORY_LOCATION=/u01/app/oraInventory
oracle.install.option=HA_CONFIG
ORACLE_BASE=/u01/app/oracle
oracle.install.asm.OSDBA=dba
oracle.install.asm.OSOPER=oper
oracle.install.asm.OSASM=asm
oracle.install.asm.SYSASMPassword="SetPWD"
oracle.install.asm.diskGroup.name=DATA
oracle.install.asm.diskGroup.redundancy=EXTERNAL
oracle.install.asm.diskGroup.AUSize=4
oracle.install.asm.diskGroup.disks=/dev/mapper/ora_01_data*
oracle.install.asm.diskGroup.diskDiscoveryString=/dev/mapper/ora_01_
data_01,/dev/mapper/ora_01_data_02,/dev/mapper/ora_01_data_03,/dev/m
apper/ora_01_data_04
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. Validate ASM filter driver status.

```

[oracle@ip-172-30-15-58 grid]$ export
ORACLE_HOME=/u01/app/oracle/product/19.0.0/grid
[oracle@ip-172-30-15-58 grid]$ export ORACLE_SID=+ASM
[oracle@ip-172-30-15-58 grid]$ export PATH=$PATH:$ORACLE_HOME/bin
[oracle@ip-172-30-15-58 grid]$ asmcmd
ASMCMD> lsdg
State      Type      Rebal   Sector   Logical_Sector   Block       AU
Total_MB   Free_MB   Req_mir_free_MB   Usable_file_MB   Offline_disks
Voting_files   Name
MOUNTED    EXTERN    N           512           512     4096   1048576
81920      81847          0           81847           0
N  DATA/
MOUNTED    EXTERN    N           512           512     4096   1048576
81920      81853          0           81853           0
N  LOGS/
ASMCMD> afd_state
ASMCMD-9526: The AFD state is 'LOADED' and filtering is 'ENABLED' on
host 'ip-172-30-15-58.ec2.internal'

```

Oracle database installation

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

```
unset ORACLE_HOME  
unset ORACLE_SID
```

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

```
mkdir /u01/app/oracle/product/19.0.0/db1  
cd /u01/app/oracle/product/19.0.0/db1
```

3. Unzip the Oracle DB installation files.

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

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

```
rm -rf OPatch
```

5. From DB home, 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 software-only installation.

```
/u01/app/oracle/product/19.0.0/db1/root.sh
```

11. As Oracle user, create the `dbca.rsp` file with the following entries:

```
gdbName=db1.demo.netapp.com
sid=db1
createAsContainerDatabase=true
numberOfPDBs=3
pdbName=db1_pdb
useLocalUndoForPDBs=true
pdbAdminPassword="yourPWD"
templateName=General_Purpose.dbc
sysPassword="yourPWD"
systemPassword="yourPWD"
dbsnmpPassword="yourPWD"
storageType=ASM
diskGroupName=DATA
characterSet=AL32UTF8
nationalCharacterSet=AL16UTF16
listeners=LISTENER
databaseType=MULTIPURPOSE
automaticMemoryManagement=false
totalMemory=8192
```

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    ip-172-30-15-58   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-8225-4784bc9843&sc_channel=ps&s_kwcid=AL!4422!3!467723097970!e!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARIaJzSrdqwQrghn6I71jiWzSeaT9Uh1-vY-VfhJixFxnv5rWwn2S7RqZOTQ0aAh7eEALw_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.

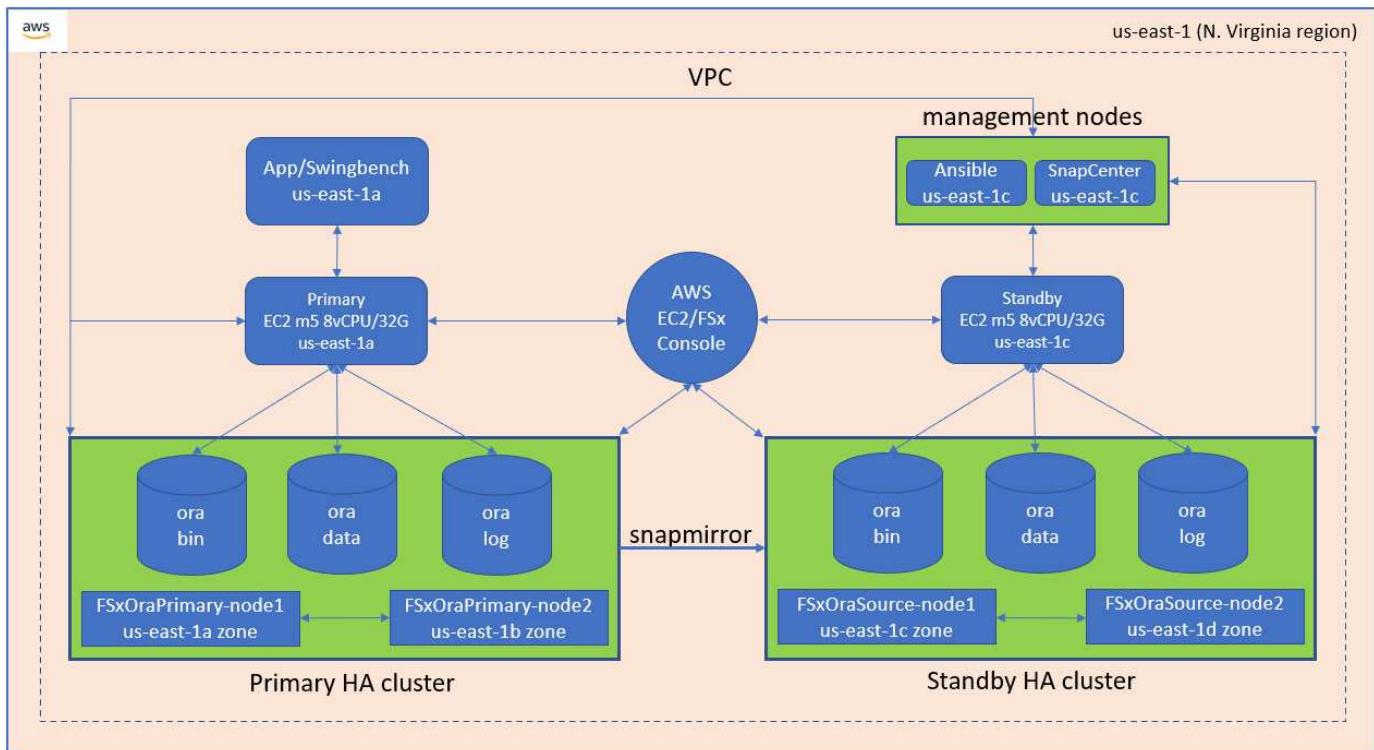
Solution architecture

[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 10Gbps 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

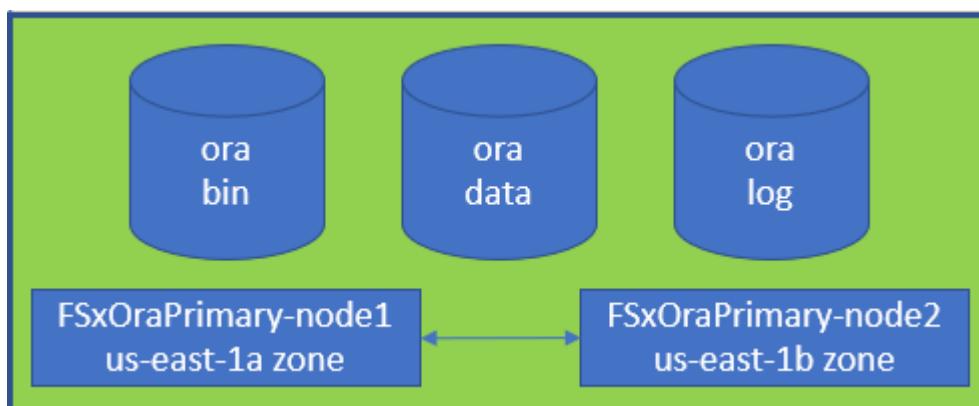
Selecting the right VM size is important for optimal performance of a relational database in a public cloud. For better performance, NetApp recommends using an EC2 M5 Series instance for Oracle deployment, which is optimized for database workloads. The same instance type is also used to power a RDS instance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

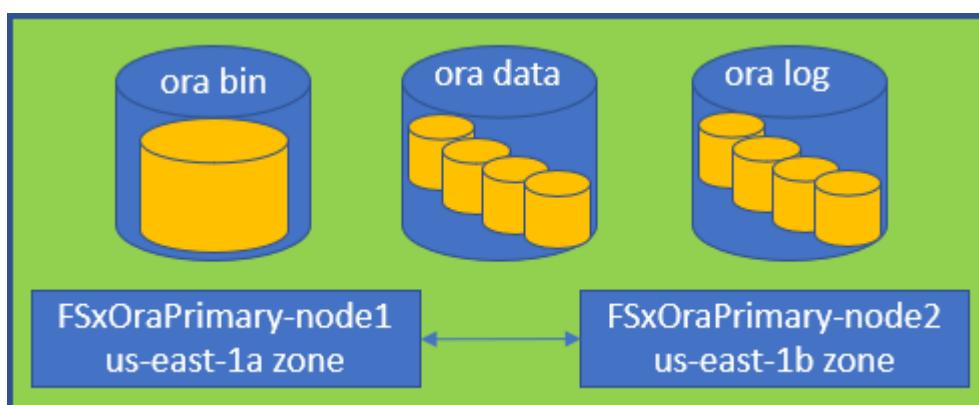
Storage layout and settings

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
<ul style="list-style-type: none"> Control files Data files Redo logs 	<code>rw, bg, hard, vers=3, proto=tcp, timeo=600, rsize=65536, wsize=65536</code>
<ul style="list-style-type: none"> ORACLE_HOME ORACLE_BASE 	<code>rw, bg, hard, vers=3, proto=tcp, timeo=600, rsize=65536, wsize=65536</code>

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.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like 'New EC2 Experience', 'EC2 Dashboard', 'Instances', 'Images', 'Elastic Block Store', and 'Network & Security'. The main area has a 'Resources' section with tables for Instances (running), Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, and Snapshots. Below this is a callout for Microsoft SQL Server Always On availability groups. To the right, there's an 'Account attributes' section with links for Supported platforms, Default VPC, Settings (which is highlighted in yellow), EBS encryption, Zones, EC2 Serial Console, Default credit specification, and Console experiments. At the bottom, there's an 'Explore AWS' section with links for 10 Things You Can Do Today to Reduce AWS Costs, Enable Best Price-Performance with AWS Graviton2, Save Up to 45% on ML Inference, and Additional information.

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.

Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	6	Dedicated Hosts	0	Elastic IPs	5
Instances	12	Key pairs	48	Load balancers	0
Placement groups	25	Security groups	33	Snapshots	0
Volumes	19				

Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Service health
[AWS Health Dashboard](#)

Explore AWS
Save up to 90% on EC2 with Spot Instances
Optimize price-performance by combining EC2 purchase options in a single EC2 ASG. [Learn more](#)

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

Step 1: Choose an Amazon Machine Image (AMI)

Are you launching a database instance? Try Amazon RDS.

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale your database on AWS by automating time-consuming database management tasks. With RDS, you can easily deploy Amazon Aurora, MariaDB, MySQL, Oracle, PostgreSQL, and SQL Server databases on AWS. Aurora is a MySQL- and PostgreSQL-compatible, enterprise-class database at 1/10th the cost of commercial databases. Learn more about RDS.

Red Hat **Select**
Free tier eligible

Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0b0af3577fe5e3532 (64-bit x86) / ami-01fc429821bf1f4b4 (64-bit Arm)

SUSE Linux **Select**
Free tier eligible

SUSE Linux Enterprise Server 15 SP3 (HVM), SSD Volume Type - ami-08895422b5f3aa64a (64-bit x86) / ami-08f182b25f271ef79 (64-bit Arm)

- 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."

Step 2: Choose an Instance Type

<input type="checkbox"/>	m4	m4.16xlarge	64	256	EBS only	Yes	25 Gigabit	Yes
<input type="checkbox"/>	m5	m5.large	2	8	EBS only	Yes	Up to 10 Gigabit	Yes
<input type="checkbox"/>	m5	m5.xlarge	4	16	EBS only	Yes	Up to 10 Gigabit	Yes
<input checked="" type="checkbox"/>	m5	m5.2xlarge	8	32	EBS only	Yes	Up to 10 Gigabit	Yes
<input type="checkbox"/>	m5	m5.4xlarge	16	64	EBS only	Yes	Up to 10 Gigabit	Yes
<input type="checkbox"/>	m5	m5.8xlarge	32	128	EBS only	Yes	10 Gigabit	Yes
<input type="checkbox"/>	m5	m5.12xlarge	48	192	EBS only	Yes	10 Gigabit	Yes
<input type="checkbox"/>	m5	m5.16xlarge	64	256	EBS only	Yes	20 Gigabit	Yes
<input type="checkbox"/>	m5	m5.24xlarge	96	384	EBS only	Yes	25 Gigabit	Yes
<input type="checkbox"/>	m5	m5.metal	96	384	EBS only	Yes	25 Gigabit	Yes

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

Screenshot of the AWS EC2 instance creation wizard Step 3: Configure Instance Details.

No default VPC found. Select another VPC, or create a new default VPC.

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1

Purchasing option: Request Spot Instances

Network: No default VPC found. Create a new default VPC.

Subnet: 250 IP Addresses available

Auto-assign Public IP: Enable

Hostname type:

DNS Hostname:

- Enable IP name IPv4 (A record) DNS requests
- Enable resource-based IPv4 (A record) DNS requests
- Enable resource-based IPv6 (AAAA record) DNS requests

Placement group: Add instance to placement group

Capacity Reservation:

Domain join directory:

IAM role:

Buttons: Cancel, Previous, **Review and Launch**, 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.

Screenshot of the AWS EC2 instance creation wizard Step 4: Add Storage.

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-03a3ad00558b4d17c	50	General Purpose SSD (gp2)	150 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

Shared file systems

You currently don't have any file systems on this instance. Select "Add file system" button below to add a file system.

Add file system

Buttons: Cancel, Previous, **Review and Launch**, Next: Add Tags

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

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more about tagging your Amazon EC2 resources.](#)

Key	(128 characters maximum)	Value	(256 characters maximum)
Instances (1) Volumes (1) Network Interfaces (1)			

This resource currently has no tags

Choose the Add tag button or click to add a Name tag.
Make sure your IAM policy includes permissions to create tags.

Add Tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

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

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security Group ID	Name	Description	Actions
<input type="checkbox"/> sg-0d746a0908b897c48	AviOccm03112021OCCM1635951256631-OCCMSecurityGroup-B3QFHUHJRUWV	NetApp OCCM Instance External Security Group	Copy to new
<input type="checkbox"/> sg-07b0625cd54aae16	AviOCCM0311OCCM1635943382952-OCCMSecurityGroup-1L8D4QX2SC945	NetApp OCCM Instance External Security Group	Copy to new
<input type="checkbox"/> sg-0618122caef6c50e9	AviOccm1103OCCM163594422113-OCCMSecurityGroup-DX5PHX6CKVKC	NetApp OCCM Instance External Security Group	Copy to new
<input type="checkbox"/> sg-0d63ea8c79897e666	AviOccm1209OCCM1631452667252-OCCMSecurityGroup-T5KVZ1Q4SH48	NetApp OCCM Instance External Security Group	Copy to new
<input type="checkbox"/> sg-0aed9f836b48c52d	AviOccmFSxOCCM1638110371156-OCCMSecurityGroup-N0ENZJW3TVYB	NetApp OCCM Instance External Security Group	Copy to new
<input type="checkbox"/> sg-083a6ea5ca9a12375	connector01OCCM1631455604110-OCCMSecurityGroup-1790QV45PH3ZW	NetApp OCCM Instance External Security Group	Copy to new
<input checked="" type="checkbox"/> sg-08148ca915189ac87	default	default VPC security group	Copy to new
<input type="checkbox"/> sg-07fc527620e3bb22	fsx02OCCM163339531669-OCCMSecurityGroup-1XZYC5WM15NP7	NetApp OCCM Instance External Security Group	Copy to new
<input type="checkbox"/> sg-0f359d2ba38db749f	SG-Version10-0CE6MEs-NetAppExternalSecurityGroup-N8B50KGTK8U	ONTAP Cloud firewall rules for management and data interface	Copy to new

Inbound rules for sg-08148ca915189ac87 (Selected security groups: sg-08148ca915189ac87)

Type (1)	Protocol (1)	Port Range (1)	Source (1)	Description (1)
All traffic	All	All	192.168.1.0/24	
All traffic	All	All	sg-08148ca915189ac87 (default)	

Cancel Previous Review and Launch

- 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.

Screenshot of the AWS EC2 Instance Launch Wizard Step 7: Review Instance Launch. The page shows the selected AMI (Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0b0af3577fe5e3532), Instance Type (m5.2xlarge), and Security Group (default). It also lists inbound rules for the security group. A modal window titled "Select an existing key pair or create a new key pair" is displayed, containing instructions about key pairs, a note about selecting a key pair, a dropdown for choosing a key pair, a dropdown for selecting a key pair, and a checkbox for acknowledging access to the private key file. Buttons for "Cancel" and "Launch Instances" are at the bottom.

Step 7: Review Instance Launch
 Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0b0af3577fe5e3532
 Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type
 Free tier eligible Root Device Type: ebs Virtualization type: hvm

Edit AMI

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
m5.2xlarge	-	8	32	EBS only	Yes	Up to 10 Gigabit

Edit instance type

Security Groups

Security Group ID	Name	Description
sg-08148ca915189ac87	default	default VPC security group

All selected security groups inbound rules

Type	Protocol	Port Range	Source	Description
All traffic	All	All	192.168.1.0/24	
All traffic	All	All	sg-08148ca915189ac87 (default)	

Edit security groups

Instance Details

Edit instance details

Storage

Edit storage

Cancel **Previous** **Launch**

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair

Select a key pair

accessstkey | RSA

I acknowledge that I have access to the corresponding private key file, and that without this file, I won't be able to log into my instance.

Cancel **Launch Instances**

- 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.

The screenshot shows the AWS FSx console interface. On the left, there's a sidebar with navigation links: Services, Resource Groups & Tag Editor, Amazon FSx, File systems, Volumes, Backups, ONTAP, Storage virtual machines, OpenZFS, Snapshots, Windows File Server, Lustre, Data repository tasks, and FSx on Service Quotas. The main content area has a title bar 'fsx (svm-005c6edf027866ca4)' with 'Delete' and 'Update' buttons. Below the title is a 'Summary' section with fields like SVM ID, SVM name, UUID, File system ID, and Resource ARN. Underneath is an 'Endpoints' section with tables for Management DNS name, NFS DNS name, and iSCSI DNS name, each with their respective IP addresses highlighted with red boxes.

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.

Select file system type

File system options

- Amazon FSx for NetApp ONTAP
- Amazon FSx for OpenZFS
- Amazon FSx for Windows File Server
- Amazon FSx for Lustre

Amazon FSx for NetApp ONTAP

Amazon FSx for NetApp ONTAP provides feature-rich, high-performance, and highly-reliable storage built on NetApp's popular ONTAP file system and fully managed by AWS.

- Broadly accessible from Linux, Windows, and macOS compute instances and containers (running on AWS or on-premises) via industry-standard NFS, SMB, and iSCSI protocols.
- Provides ONTAP's popular data management capabilities like Snapshots, SnapMirror (for data replication), FlexClone (for data cloning), and data compression / deduplication.
- Delivers hundreds of thousands of IOPS with consistent sub-millisecond latencies, and up to 3 GB/s of throughput.
- Offers highly-available and highly-durable multi-AZ SSD storage with support for cross-region replication and built-in, fully managed backups.
- Automatically tiers infrequently-accessed data to capacity pool storage, a fully elastic storage tier that can scale to petabytes in size and is cost-optimized for infrequently-accessed data.
- Integrates with Microsoft Active Directory (AD) to support Windows-based environments and enterprises.

Cancel

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.

File system details

File system name - optional [Info](#)

aws_ora_prod

Maximum of 256 Unicode letters, whitespace, and numbers, plus + - = . _ : /

Deployment type [Info](#)

Multi-AZ

Single-AZ

SSD storage capacity [Info](#)

1024

Minimum 1024 GiB; Maximum 192 TiB.

Provisioned SSD IOPS

Amazon FSx provides 3 IOPS per GiB of storage capacity. You can also provision additional SSD IOPS as needed.

Automatic (3 IOPS per GiB of SSD storage)

User-provisioned

40000

Maximum 80,000 IOPS

Throughput capacity [Info](#)

The sustained speed at which the file server hosting your file system can serve data. The file server can also burst to higher speeds for periods of time.

Recommended throughput capacity

128 MB/s

Specify throughput capacity

Throughput capacity

512 MB/s



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.

Network & security

Virtual Private Cloud (VPC) [Info](#)

Specify the VPC from which your file system is accessible.

vpc-0474064fc537e5182



VPC Security Groups [Info](#)

Specify VPC Security Groups to associate with your file system's network interfaces.

Choose VPC security group(s)



sg-08148ca915189ac87 (default)

Preferred subnet [Info](#)

Specify the preferred subnet for your file system.

subnet-08c952541f4ab282d (us-east-1a)



Standby subnet

subnet-0a84d6eeeb0f4e5c0 (us-east-1b)



VPC route tables

Specify the VPC route tables associated with your file system.

VPC's default route table

Select one or more VPC route tables

Endpoint IP address range

Specify the IP address range in which the endpoints to access your file system will be created

No preference

Select an IP address range

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

Security & encryption

Encryption key [Info](#)

AWS Key Management Service (KMS) encryption key that protects your file system data at rest.

aws/fsx (default)



Description	Account	KMS key ID
Default master key that protects my FSx resources when no other key is defined	759995470648	5b31feff-6759-4306-a852-9c99a743982a

File system administrative password

Password for this file system's "fsxadmin" user, which you can use to access the ONTAP CLI or REST API.

Don't specify a password

Specify a password

Password

Confirm password

6. Enter the SVM name and the vsadmin password.

Default storage virtual machine configuration

Storage virtual machine name

SVM administrative password
Password for this SVM's "vsadmin" user, which you can use to access the ONTAP CLI or REST API.

Don't specify a password
 Specify a password
Password

Confirm password

Active Directory
Joining an Active Directory enables access from Windows and MacOS clients over the SMB protocol.

Do not join an Active Directory
 Join an Active Directory

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

Default volume configuration

Volume name

vol1

Maximum of 203 alphanumeric characters, plus _.

Junction path

/vol1

The location within your file system where your volume will be mounted.

Volume size

1024

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.

Auto



► Backup and maintenance - *optional*

► Tags - *optional*

Cancel

Back

Next

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

Screenshot of the AWS FSx Create file system wizard Step 3: Review and create. The summary table shows the following configuration:

Attribute	Value	Editable after creation
File system type	Amazon FSx for NetApp ONTAP	
File system name	aws_ora_prod	<input checked="" type="checkbox"/>
Deployment type	Multi-AZ	
Storage type	SSD	
SSD storage capacity	1,024 GiB	<input checked="" type="checkbox"/>
Minimum SSD IOPS	40000 IOPS	<input checked="" type="checkbox"/>
Throughput capacity	512 MB/s	<input checked="" type="checkbox"/>
Virtual Private Cloud (VPC)	vpc-0474064fc537e5182	
VPC Security Groups	sg-08148ca915189ac87	<input checked="" type="checkbox"/>
Preferred subnet	subnet-08c952541f4ab282d	
Standby subnet	subnet-0a84d6eeeb0f4e5c0	
VPC route tables	VPC's default route table	
Endpoint IP address range	No preference	
KMS key ID	arn:aws:kms:us-east-1:759995470648:key/5b31feff-6759-4306-a852-9c99a743982a	
Daily automatic backup window	No preference	<input checked="" type="checkbox"/>
Automatic backup	7 day(s)	<input checked="" type="checkbox"/>

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

X

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

X

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

Create volume

File system

ONTAP | fs-0a51a3f08922224d5 | rdscustomfs007

Storage virtual machine

svm-005c6edf027866ca4 | fsx

Volume name

db1_log

Maximum of 203 alphanumeric characters, plus _.

Junction path

/db1_log

The location within your file system where your volume will be mounted.

Volume size

256000

Minimum 20 MiB; Maximum 104857600 MiB

Storage efficiency

Select whether you would like to enable ONTAP storage efficiencies on your volume: deduplication, compression, and compaction.

Enabled (recommended)

Disabled

Capacity pool tiering policy

You can optionally enable automatic tiering of your data to lower-cost capacity pool storage.

None

Cancel **Confirm**



Creating iSCSI LUNs is not currently supported by the FSx console. For iSCSI LUNs deployment for Oracle, the volumes and LUNs can be created by using automation for ONTAP with the NetApp Automation Toolkit.

Install and configure Oracle on an EC2 instance with FSx database volumes

The NetApp automation team provide an automation kit to run Oracle installation and configuration on EC2 instances according to best practices. The current version of the automation kit supports Oracle 19c on NFS with the default RU patch 19.8. The automation kit can be easily adapted for other RU patches if needed.

Prepare a Ansible controller to run automation

Follow the instruction in the section "[Creating and connecting to an EC2 instance for hosting Oracle database](#)" to provision a small EC2 Linux instance to run the Ansible controller. Rather than using RedHat, Amazon Linux t2.large with 2vCPU and 8G RAM should be sufficient.

Retrieve NetApp Oracle deployment automation toolkit

Log into the EC2 Ansible controller instance provisioned from step 1 as ec2-user and from the ec2-user home directory, execute the `git clone` command to clone a copy of the automation code.

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

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

Execute automated Oracle 19c deployment using automation toolkit

See these detailed instruction [CLI deployment Oracle 19c Database](#) to deploy Oracle 19c with CLI automation. There is a small change in command syntax for playbook execution because you are using an SSH key pair instead of a password for host access authentication. The following list is a high level summary:

1. By default, an EC2 instance uses an SSH key pair for access authentication. From Ansible controller automation root directories `/home/ec2-user/na_oracle19c_deploy`, and `/home/ec2-user/na_rds_fsx_oranfs_config`, make a copy of the SSH key `accesststkey.pem` for the Oracle host deployed in the step "[Creating and connecting to an EC2 instance for hosting Oracle database](#)".
2. Log into the EC2 instance DB host as ec2-user, and install the python3 library.

```
sudo yum install python3
```

3. Create a 16G swap space from the root disk drive. By default, an EC2 instance creates zero swap space. Follow this AWS documentation: [How do I allocate memory to work as swap space in an Amazon EC2 instance by using a swap file?](#).
4. Return to the Ansible controller (`cd /home/ec2-user/na_rds_fsx_oranfs_config`), and execute the preclone playbook with the appropriate requirements and `linux_config` tags.

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

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

5. Switch to the `/home/ec2-user/na_oracle19c_deploy-master` directory, read the `README` file, and populate the `global_vars.yml` file with the relevant global parameters.

6. Populate the `host_name.yml` file with the relevant parameters in the `host_vars` directory.

7. Execute the playbook for Linux, and press Enter when prompted for the `vsadmin` password.

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

8. Execute the playbook for Oracle, and press enter when prompted for the `vsadmin` password.

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

Change the permission bit on the SSH key file to 400 if needed. Change the Oracle host (`ansible_host` in the `host_vars` file) IP address to your EC2 instance public address.

Setting up SnapMirror between primary and standby FSx HA cluster

For high availability and disaster recovery, you can set up SnapMirror replication between the primary and standby FSx storage cluster. Unlike other cloud storage services, FSx enables a user to control and manage storage replication at a desired frequency and replication throughput. It also enables users to test HA/DR without any effect on availability.

The following steps show how to set up replication between a primary and standby FSx storage cluster.

1. Setup primary and standby cluster peering. Log into the primary cluster as the `fsxadmin` user and execute the following command. This reciprocal create process executes the `create` command on both the primary cluster and the standby cluster. Replace `standby_cluster_name` with the appropriate name for your environment.

```
cluster peer create -peer-addrs  
standby_cluster_name,inter_cluster_ip_address -username fsxadmin  
-initial-allowed-vserver-peers *
```

2. Set up vServer peering between the primary and standby cluster. Log into the primary cluster as the `vsadmin` user and execute the following command. Replace `primary_vserver_name`, `standby_vserver_name`, `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_FsxEraSource
      svm_FsxEraTarget
                  peered      FsxId0b6a95149d07aa82e
                                         snapmirror      svm_FsxEraTarget

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.

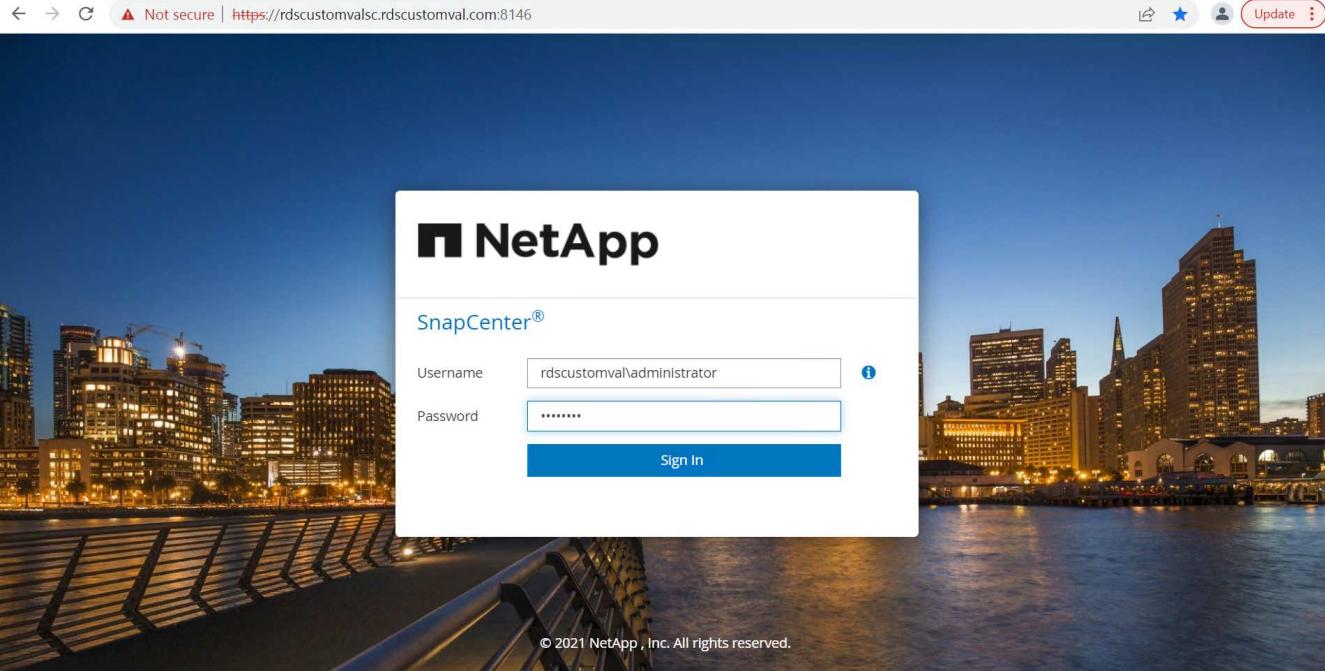
SnapCenter Deployment

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.



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

NetApp SnapCenter®

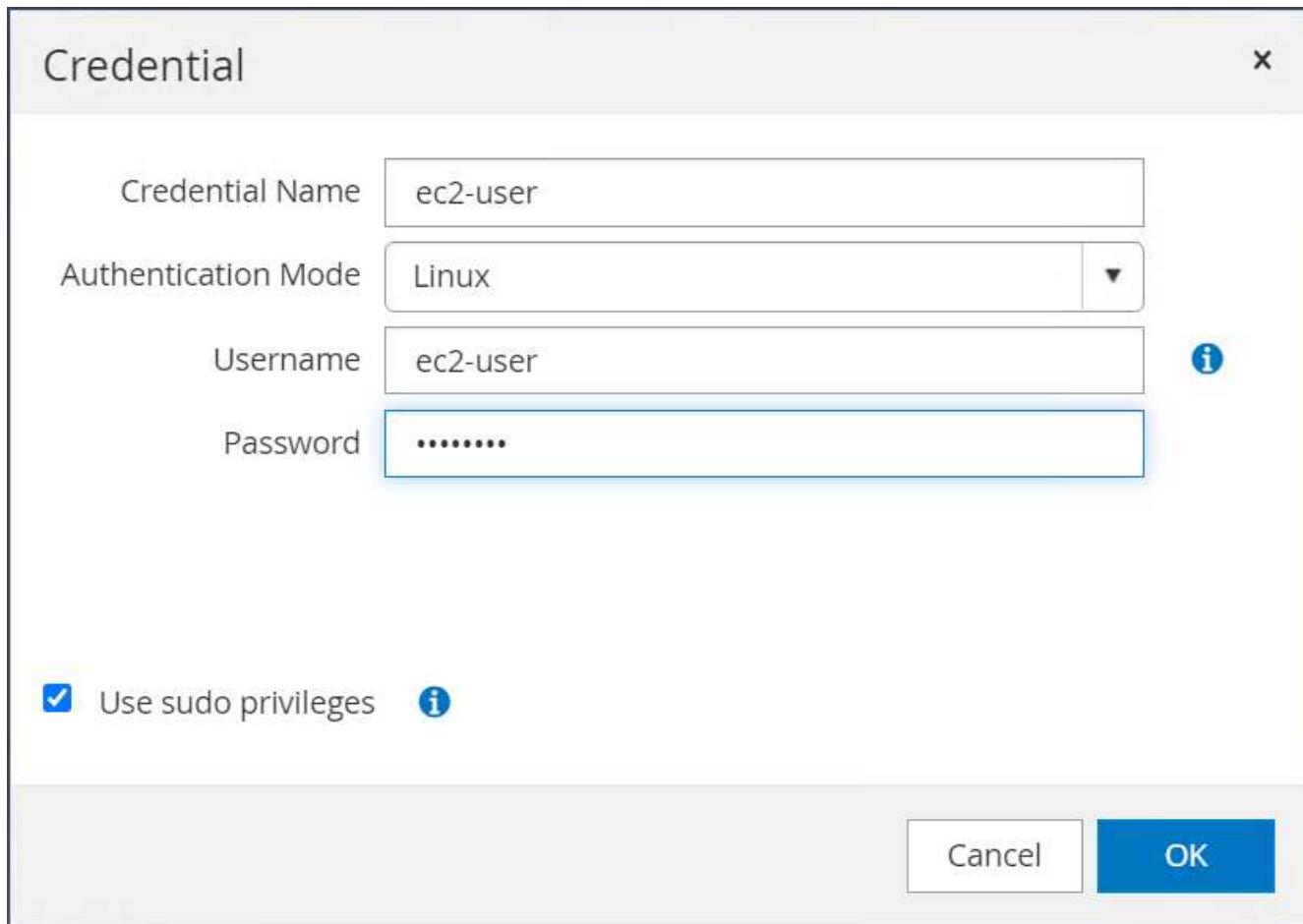
Global Settings Policies Users and Access Roles **Credential** Software

Search by Credential Name

	Credential Name	Authentication Mode	Details
	244rdscustomdb	SQL	UserId:admin
	42rdscustomdb	SQL	UserId:admin
	admin	SQL	UserId:admin
	administrator	Windows	UserId:administrator
	ec2-user	Linux	UserId:ec2-user
	onpremSQL	Windows	UserId:rdscustomval\administrator
	rdsdb2	Windows	UserId:administrator
	rdsdb244	Windows	UserId:administrator
	rdssql	Windows	UserId:administrator
	tst244	SQL	UserId:admin
	tstcredfordemo	Windows	UserId:administrator

New **Modify** **Delete**

- Reset the ec2-user password and enable password SSH authentication by editing the `/etc/ssh/sshd_config` file on the EC2 instance host.
- 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.

The screenshot shows the NetApp SnapCenter interface. The left sidebar has a 'Hosts' icon highlighted. The main area is titled 'Managed Hosts' and lists two hosts:

	Name	Type	System	Plug-in	Version	Overall Status
<input type="checkbox"/>	RDSAMA7-VJ0DQK0	Windows	Stand-alone	Microsoft Windows Server, Microsoft SQL Server	4.5	Host down
<input type="checkbox"/>	rdscustommssql1.rdscustomval.com	Windows	Stand-alone	Microsoft Windows Server, Microsoft SQL Server	4.5	Running

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

The screenshot shows the 'Add Host' dialog box. It includes fields for Host Type (Linux), Host Name (10.0.0.151), and Credentials (ec2-user). Below the form is a section for selecting plug-ins:

Select Plug-ins to Install SnapCenter Plug-Ins Package 4.5 P2 for Linux

- Oracle Database
- SAP HANA

[More Options](#): Port, Install Path, Custom Plug-Ins...

At the bottom are 'Submit' and 'Cancel' buttons.

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

More Options

Port	8145	
Installation Path	/opt/NetApp/snapcenter	
<input checked="" type="checkbox"/> Skip preinstall checks		
Custom Plug-ins Choose a File <input type="button" value="Browse"/> <input type="button" value="Upload"/> <div style="border: 1px solid #ccc; padding: 5px; height: 40px; margin-top: 10px;">No plug-ins found.</div>		
<input type="button" value="Save"/> <input type="button" value="Cancel"/>		

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

Confirm Fingerprint

Authenticity of the host cannot be determined		
Host name	Fingerprint	Valid
ip-10-0-0-151.ec2.internal	ssh-rsa 2048 97:6F:3C:7D:38:42:F6:54:B7:AF:E3:61:61:BA:2E:6F	
<input type="button" value="Confirm and Submit"/> <input type="button" value="Close"/>		

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

Managed Hosts							
Search by Name		Add		Remove		Refresh	
	Name	Type	System	Plug-in	Version	Overall Status	
<input type="checkbox"/>	ip-10-0-0-151.ec2.internal	Linux	Stand-alone	UNIX, Oracle Database	4.5		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.

The screenshot shows the NetApp SnapCenter interface. On the left, there's a sidebar with icons for Dashboard, Resources (selected), Monitor, Reports, Hosts, Storage Systems, Settings, and Alerts. The main area has tabs for View (Database selected) and Search databases. A table lists Oracle databases: ORCL (Single Instance, Host/Cluster: ip-10-0-0-151.ec2.internal). The Resource Group column shows two groups: 'orc1_full_bkup' and 'orc1_log_bkup'. The Policies column indicates 'Oracle full backup' and 'Oracle log backup'. The Last Backup column shows '03/24/2022 8:40:08 PM' with a refresh icon. The Overall Status column shows 'Backup succeeded'. At the bottom right of the table are 'Refresh Resources' and 'New Resource Group' buttons.

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 fallback 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.

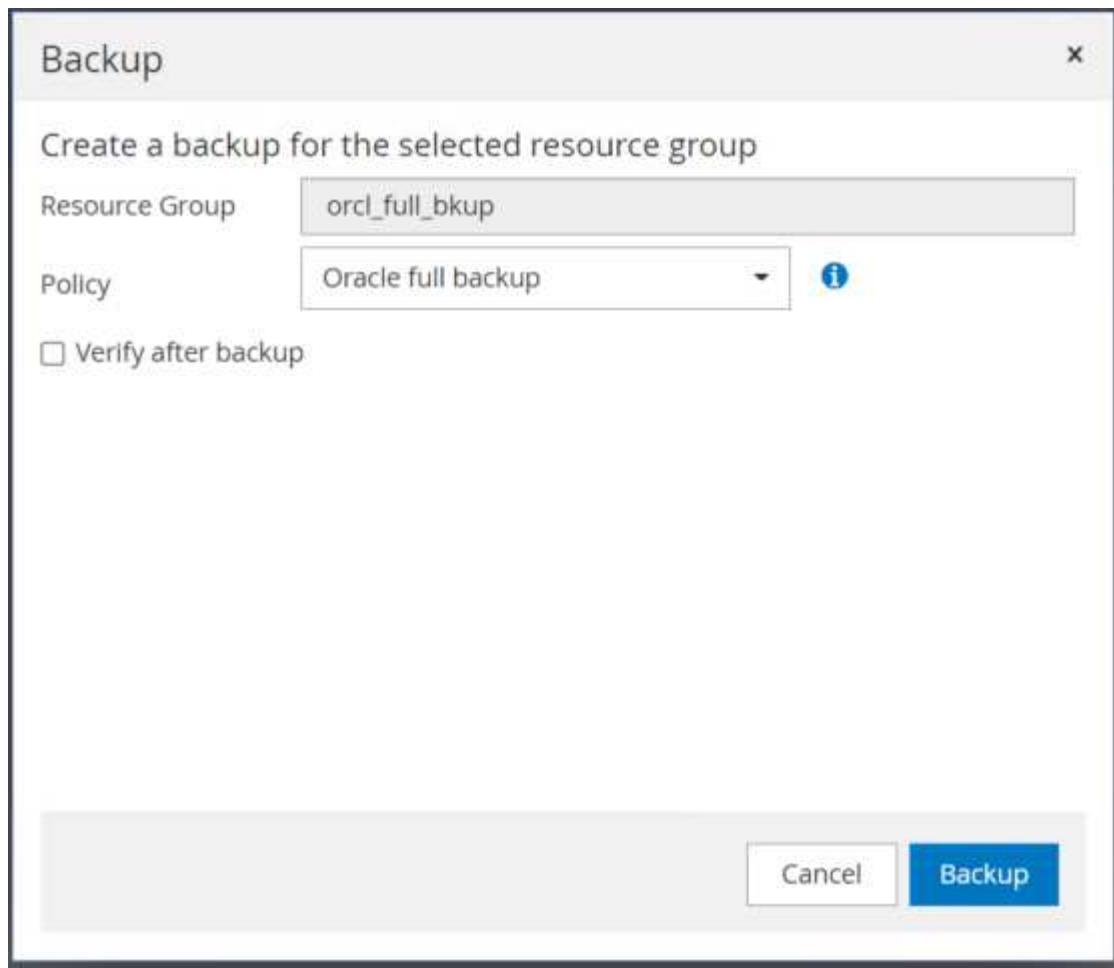
1. Log into the SnapCenter UI and click Resources in the left-side menu. From the View dropdown, change to the Resource Group view.

Name	Resources	Tags	Policies
ordl_full_bkup	1	ora_fullbkup	Oracle full backup
ordl_log_bkup	1	ora_logbkup	Oracle log backup

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

Name	Resource Name	Type	Host
ordl_full_bkup	ORCL	Oracle Database	ip-10-0-0-151.ec2.internal

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.

Backup Name	Count	Type	End Date	Verified	Mounted	RMAN Cataloged	SCN
[p-10-0-0-151_03-25-2022_00.34.20.4541.1]	1	Log	03/25/2022 12:34:37 AM	Not Applicable	False	Not Cataloged	1733264
[p-10-0-0-151_03-25-2022_00.34.20.4541.0]	1	Data	03/25/2022 12:34:31 AM	Unverified	False	Not Cataloged	1733220

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.

The screenshot shows the NetApp SnapCenter interface. On the left is a navigation sidebar with icons for Dashboard, Resources (selected), Monitor, Reports, Hosts, Storage Systems, Settings, and Alerts. The main area has a dropdown menu set to 'Oracle Database'. Below it, a 'View' dropdown is set to 'Resource Group' and a search bar says 'Search resource group'. A table lists two resources: 'ordl_full_bkup' and 'ordl_log_bkup'. Both have a count of 1, a tag 'ora_fullbkup' or 'ora_logbkup', and a policy 'Oracle full backup' or 'Oracle log backup' respectively.

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

This screenshot shows the 'ordl_log_bkup Details' page. The left sidebar shows the resource group 'ordl_log_bkup' is selected. The main table shows one resource named 'ordl_log_bkup' with type 'ORCL' and host 'ip-10-0-0-151.ec2.internal'. On the right, there are buttons for 'Modify Resource Group', 'Backup Now' (highlighted in yellow), 'Maintenance', and 'Delete'.

3. Click Backup and then confirm the backup to start an archive log backup.

A modal dialog box titled 'Backup' is shown. It asks 'Create a backup for the selected resource group'. The 'Resource Group' field contains 'ordl_log_bkup'. The 'Policy' field is set to 'Oracle log backup'. At the bottom are 'Cancel' and 'Backup' buttons.

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.

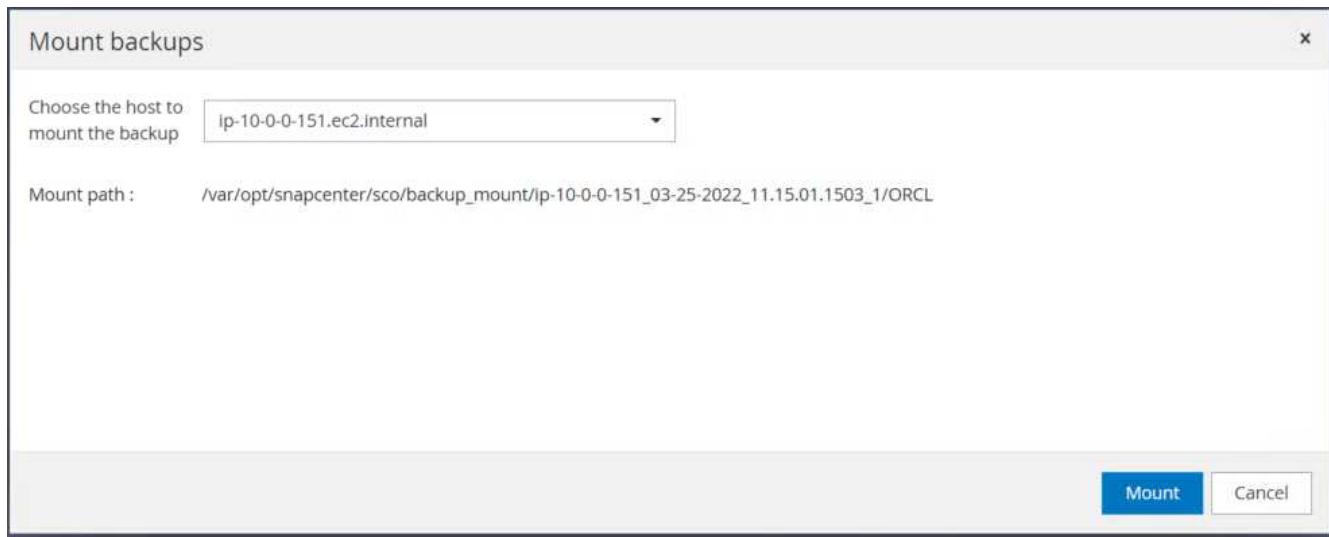
2. Select the database backup copy and the desired point in time to be restored. Also mark down the corresponding SCN number for the point in time. The point-in-time restore can be performed using either the time or the SCN.

Backup Name	Count	Type	End Date	Verified	Mounted	RMAN Cataloged	SCN
ip-10-0-0-151_03-25-2022_12.40.01.1098_1	1	Log	03/25/2022 12:40:09 PM	Not Applicable	False	Not Cataloged	1784293
ip-10-0-0-151_03-25-2022_12.25.01.0080_1	1	Log	03/25/2022 12:25:09 PM	Not Applicable	False	Not Cataloged	1783383
ip-10-0-0-151_03-25-2022_12.10.01.1097_1	1	Log	03/25/2022 12:10:09 PM	Not Applicable	False	Not Cataloged	1782417
ip-10-0-0-151_03-25-2022_11.55.01.0500_1	1	Log	03/25/2022 11:55:09 AM	Not Applicable	False	Not Cataloged	1781160
ip-10-0-0-151_03-25-2022_11.40.01.0323_1	1	Log	03/25/2022 11:40:09 AM	Not Applicable	False	Not Cataloged	1780268
ip-10-0-0-151_03-25-2022_11.25.01.0430_1	1	Log	03/25/2022 11:25:09 AM	Not Applicable	False	Not Cataloged	1779368
ip-10-0-0-151_03-25-2022_11.15.01.1503_1	1	Log	03/25/2022 11:15:17 AM	Not Applicable	False	Not Cataloged	1778546
ip-10-0-0-151_03-25-2022_11.15.01.1503_0	1	Data	03/25/2022 11:15:11 AM	Unverified	False	Not Cataloged	1778504
ip-10-0-0-151_03-25-2022_11.10.01.1834_1	1	Log	03/25/2022 11:10:09 AM	Not Applicable	False	Not Cataloged	1778184

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

Backup Name	Count	Type	End Date	Verified	Mounted	RMAN Cataloged	SCN
ip-10-0-0-151_03-25-2022_12.40.01.1098_1	1	Log	03/25/2022 12:40:09 PM	Not Applicable	False	Not Cataloged	1784293
ip-10-0-0-151_03-25-2022_12.25.01.0080_1	1	Log	03/25/2022 12:25:09 PM	Not Applicable	False	Not Cataloged	1783383
ip-10-0-0-151_03-25-2022_12.10.01.1097_1	1	Log	03/25/2022 12:10:09 PM	Not Applicable	False	Not Cataloged	1782417
ip-10-0-0-151_03-25-2022_11.55.01.0500_1	1	Log	03/25/2022 11:55:09 AM	Not Applicable	False	Not Cataloged	1781160
ip-10-0-0-151_03-25-2022_11.40.01.0323_1	1	Log	03/25/2022 11:40:09 AM	Not Applicable	False	Not Cataloged	1780268
ip-10-0-0-151_03-25-2022_11.25.01.0430_1	1	Log	03/25/2022 11:25:09 AM	Not Applicable	False	Not Cataloged	1779368
ip-10-0-0-151_03-25-2022_11.15.01.1503_1	1	Log	03/25/2022 11:15:17 AM	Not Applicable	False	Not Cataloged	1778546
ip-10-0-0-151_03-25-2022_11.15.01.1503_0	1	Data	03/25/2022 11:15:11 AM	Unverified	False	Not Cataloged	1778504
ip-10-0-0-151_03-25-2022_11.10.01.1834_1	1	Log	03/25/2022 11:10:09 AM	Not Applicable	False	Not Cataloged	1778184

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 that log volume mounted and also the mount point path.

```
[root@ip-10-0-0-151 ec2-user]# df -h
Filesystem      Size  Used Avail Mounted on
/devtmpfs        7.6G   0    7.6G  0% /dev
tmpfs           1.6G  7.0G  8.3G  46% /dev/shm
tmpfs           7.7G  604K  7.6G  1% /run
tmpfs           7.7G   0    7.7G  0% /sys/fs/cgroup
/dev/nvme0n1p1   9.8G  5.4G  4.3G  56% /
198.19.255.68:/ora_nfs_log  48G  95M  48G  1% /ora_nfs_log
198.19.255.68:/ora_nfs_data  48G  3.4G  45G  8% /ora_nfs_data
/dev/mapper/dbdata01-lvdbdata01  40G  471M  38G  2% /rdsdbdata
/dev/nvme5n1     25G   12G  13G  49% /rdsdbbin
tmpfs           1.6G   0    1.6G  0% /run/user/61001
tmpfs           1.6G   0    1.6G  0% /run/user/61005
198.19.255.68:/Scef91c793-5583-480d-9a34-6275dab17f5b  48G  91M  48G  1% /var/opt/snapcenter/sco/backup_mount/ip-10-0-0-151_03-25-2022_11.15.01.1503_1/ORCL/
[root@ip-10-0-0-151 ec2-user]#
```

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.

Manage Copies

Backup Name	Count	Type	End Date	Verified	Mounted	RMAN Cataloged	SCN
ip-10-0-0-151_03-25-2022_12.10.01.1097_1	1	Log	03/25/2022 12:10:09 PM	Not Applicable	False	Not Cataloged	1782417
ip-10-0-0-151_03-25-2022_11.55.01.0500_1	1	Log	03/25/2022 11:55:09 AM	Not Applicable	False	Not Cataloged	1781160
ip-10-0-0-151_03-25-2022_11.40.01.0323_1	1	Log	03/25/2022 11:40:09 AM	Not Applicable	False	Not Cataloged	1780268
ip-10-0-0-151_03-25-2022_11.25.01.0430_1	1	Log	03/25/2022 11:25:09 AM	Not Applicable	False	Not Cataloged	1779368
ip-10-0-0-151_03-25-2022_11.15.01.1503_1	1	Log	03/25/2022 11:15:17 AM	Not Applicable	True	Not Cataloged	1778546
ip-10-0-0-151_03-25-2022_11.15.01.1503_0	1	Data	03/25/2022 11:15:11 AM	Unverified	False	Not Cataloged	1778504
ip-10-0-0-151_03-25-2022_11.10.01.1834_1	1	Log	03/25/2022 11:10:09 AM	Not Applicable	False	Not Cataloged	1778184

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

Restore ORCL

1 Restore Scope

Restore Scope

All Datafiles

Tablespaces

Control files

Database State

Change database state if needed for restore and recovery

Restore Mode

Force in place restore

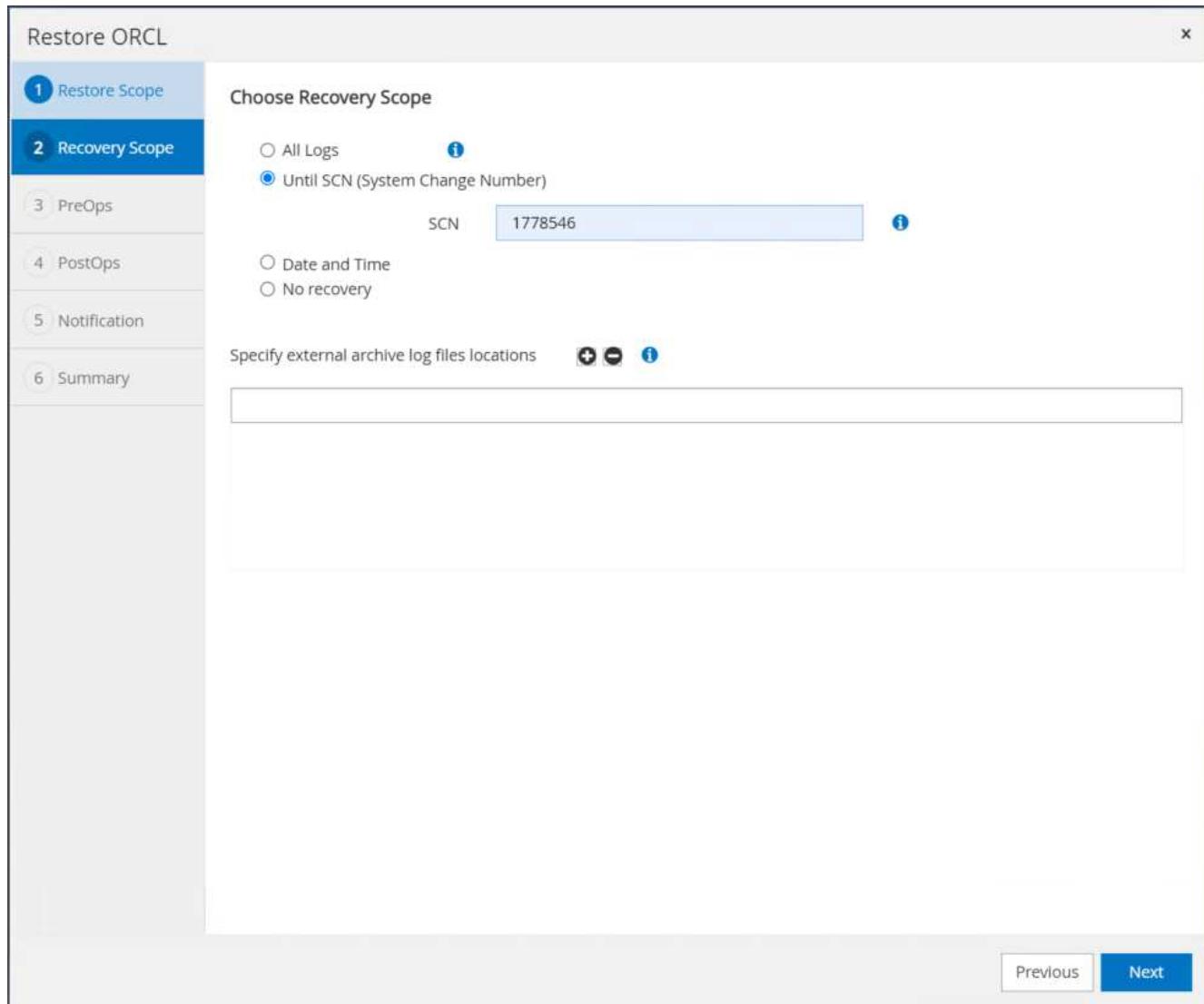
If this check box is not selected and if any of the in place restore criteria is not met, restore will be performed using the connect and copy method. The connect and copy restore method might take time based on the files being restored.

Previous

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.

Restore ORCL

1 Restore Scope

2 Recovery Scope

3 PreOps

4 PostOps

5 Notification

6 Summary

Specify optional scripts to run before performing a restore job i

Prescript full path Enter Prescript path

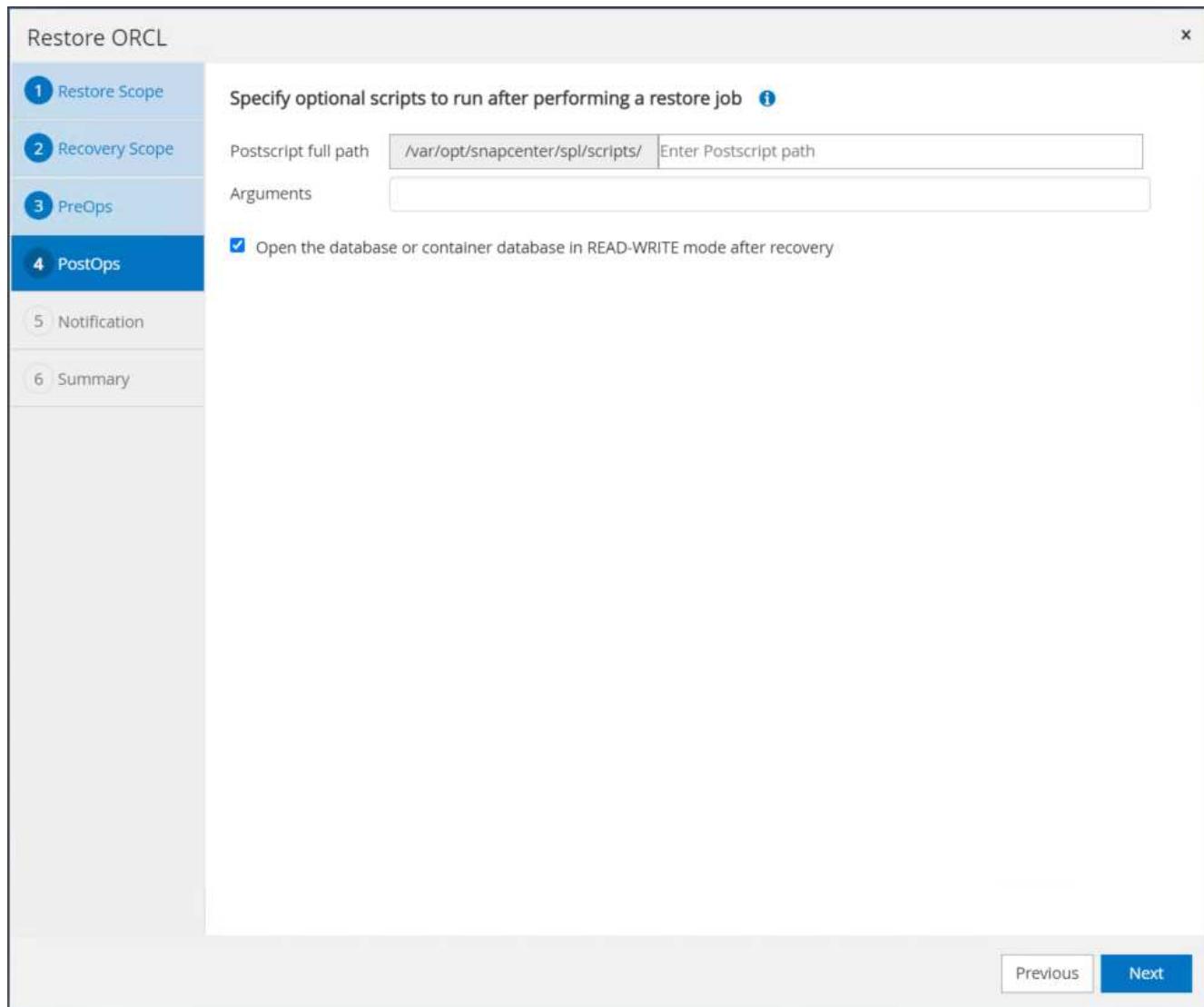
Arguments

Script timeout

Previous Next

The screenshot shows the Oracle SnapCenter Restore ORCL wizard, specifically Step 3: PreOps. The left sidebar lists steps 1 through 6. The main area is titled "Specify optional scripts to run before performing a restore job" and includes fields for "Prescript full path" (set to "/var/opt/snapcenter/spl/scripts/"), "Arguments" (empty), and "Script timeout" (set to "60 secs"). Navigation buttons "Previous" and "Next" are at the bottom right.

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.

Restore ORCL

Provide email settings i

Email preference	Never
From	From email
To	Email to
Subject	Notification

Attach job report

Previous **Next**

The screenshot shows a step-by-step restoration wizard titled "Restore ORCL". The current step is "5 Notification". On the left, a vertical navigation bar lists steps 1 through 6: "1 Restore Scope", "2 Recovery Scope", "3 PreOps", "4 PostOps", "5 Notification" (which is highlighted in blue), and "6 Summary". The main panel contains fields for providing email settings, including dropdowns for "Email preference" (set to "Never"), "From" (set to "From email"), "To" (set to "Email to"), and "Subject" (set to "Notification"). There is also an unchecked checkbox for "Attach job report". At the bottom right, there are "Previous" and "Next" buttons.

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

Restore ORCL

X

1 Restore Scope

2 Recovery Scope

3 PreOps

4 PostOps

5 Notification

6 Summary

Summary

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

Previous **Finish**

14. Validate the restore from SnapCenter.

Job Details

Restore 'ip-10-0-0-151.ec2.internal\ORCL'

- ✓ ▾ Restore 'ip-10-0-0-151.ec2.internal\ORCL'
- ✓ ▾ ip-10-0-0-151.ec2.Internal
 - ✓ ► Prescripts
 - ✓ ► Pre Restore
 - ✓ ► Restore
 - ✓ ► Post Restore
 - ✓ ► Postscripts
 - ✓ ► Post Restore Cleanup
 - ✓ ► Data Collection
 - ✓ ► Send EMS Messages

Task Name: ip-10-0-0-151.ec2.Internal Start Time: 03/25/2022 3:33:53 PM End Time: 03/25/2022 3:35:10 PM

[View Logs](#) [Cancel Job](#) [Close](#)

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.

Name	Resource Name	Type	Host
ord_full_bkup	ORCL	Oracle Database	ip-10-0-0-151.ec2.internal
ord_log_bkup			

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.

Backup Name	Count	Type	End Date	Verified	Mounted	RMAN Cataloged	SCN
ip-10-0-0-151_03-25-2022_17.55.01.0197_1	1	Log	03/25/2022 5:55:09 PM	Not Applicable	False	Not Cataloged	1789099
ip-10-0-0-151_03-25-2022_17.50.55.0853_1	1	Log	03/25/2022 5:51:12 PM	Not Applicable	False	Not Cataloged	1788879
ip-10-0-0-151_03-25-2022_17.50.55.0853_0	1	Data	03/25/2022 5:51:05 PM	Unverified	False	Not Cataloged	1788832
ip-10-0-0-151_03-25-2022_17.40.00.9758_1	1	Log	03/25/2022 5:40:08 PM	Not Applicable	False	Not Cataloged	1788110
ip-10-0-0-151_03-25-2022_17.25.01.0539_1	1	Log	03/25/2022 5:25:08 PM	Not Applicable	False	Not Cataloged	1787180

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

ORCL Topology

Manage Copies

Local copies

Primary Backup(s)						
Backup Name	Count	Type	IF	End Date	Verified	Mounted
ip-10-0-0-151_03-25-2022_18.55.01.0309_1	1	Log		03/25/2022 6:55:09 PM	Not Applicable	False
ip-10-0-0-151_03-25-2022_18.40.00.9602_1	1	Log		03/25/2022 6:40:23 PM	Not Applicable	False
ip-10-0-0-151_03-25-2022_17.55.01.0197_1	1	Log		03/25/2022 5:55:09 PM	Not Applicable	False
ip-10-0-0-151_03-25-2022_17.50.55.0853_1	1	Log		03/25/2022 5:51:12 PM	Not Applicable	False
ip-10-0-0-151_03-25-2022_17.50.55.0853_0	1	Data		03/25/2022 5:51:05 PM	Unverified	False
ip-10-0-0-151_03-25-2022_17.40.00.9758_1	1	Log		03/25/2022 5:40:08 PM	Not	False

Mount backups

Choose the host to mount the backup : ip-10-0-0-47.ec2.internal

Mount path : /var/opt/snapcenter/sco/backup_mount/ip-10-0-0-151_03-25-2022_17.50.55.0853_1/ORCL

Mount **Cancel**

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

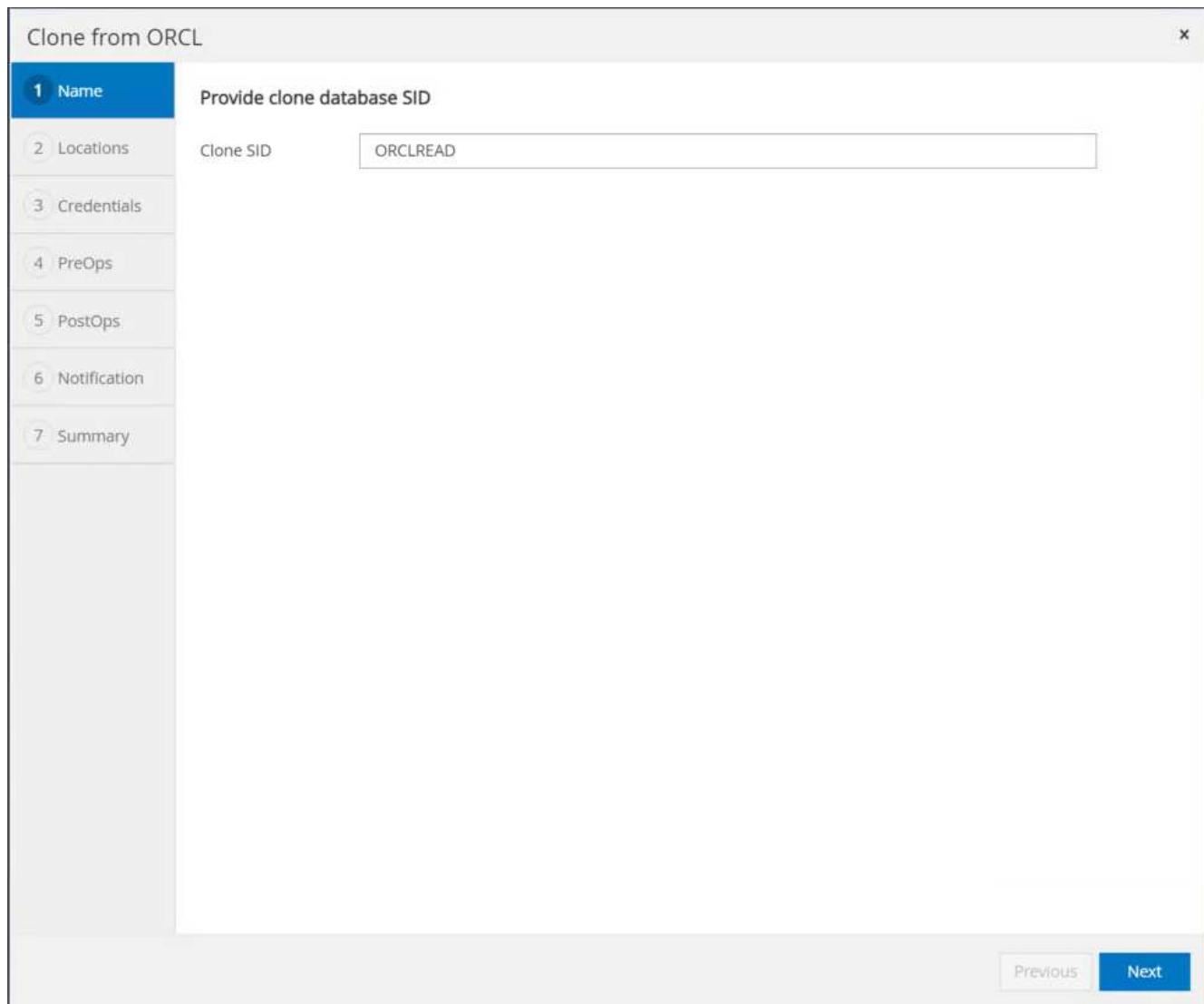
ORCL Topology

Manage Copies

Local copies

Primary Backup(s)						
Backup Name	Count	Type	IF	End Date	Verified	Mounted
ip-10-0-0-151_03-25-2022_17.55.01.0197_1	1	Log		03/25/2022 5:55:09 PM	Not Applicable	False
ip-10-0-0-151_03-25-2022_17.50.55.0853_1	1	Log		03/25/2022 5:51:12 PM	Not Applicable	False
ip-10-0-0-151_03-25-2022_17.50.55.0853_0	1	Data		03/25/2022 5:51:05 PM	Unverified	False
ip-10-0-0-151_03-25-2022_17.40.00.9758_1	1	Log		03/25/2022 5:40:08 PM	Not Applicable	False
ip-10-0-0-151_03-25-2022_17.25.01.0539_1	1	Log		03/25/2022 5:25:08 PM	Not	False

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.

Clone from ORCL

1 Name

Select the host to create a clone

Clone host: ip-10-0-0-47.ec2.internal

2 Locations

Datafile locations: /ora_nfs_data_ORCLREAD

Control files: /ora_nfs_data_ORCLREAD/ORCLREAD/control/control01.ctl

Redo logs:

Group	Size	Unit	Number of files
RedoGroup 1	128	MB	1
RedoGroup 2	128	MB	1

3 Credentials

4 PreOps

5 PostOps

6 Notification

7 Summary

Previous Next

Group	Size	Unit	Number of files
RedoGroup 1	128	MB	1
RedoGroup 2	128	MB	1

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

Clone from ORCL

1 Name

2 Locations

3 Credentials

4 PreOps

5 PostOps

6 Notification

7 Summary

Database Credentials for the clone

Credential name for sys user - + ⓘ

Database port

Oracle Home Settings ⓘ

Oracle Home

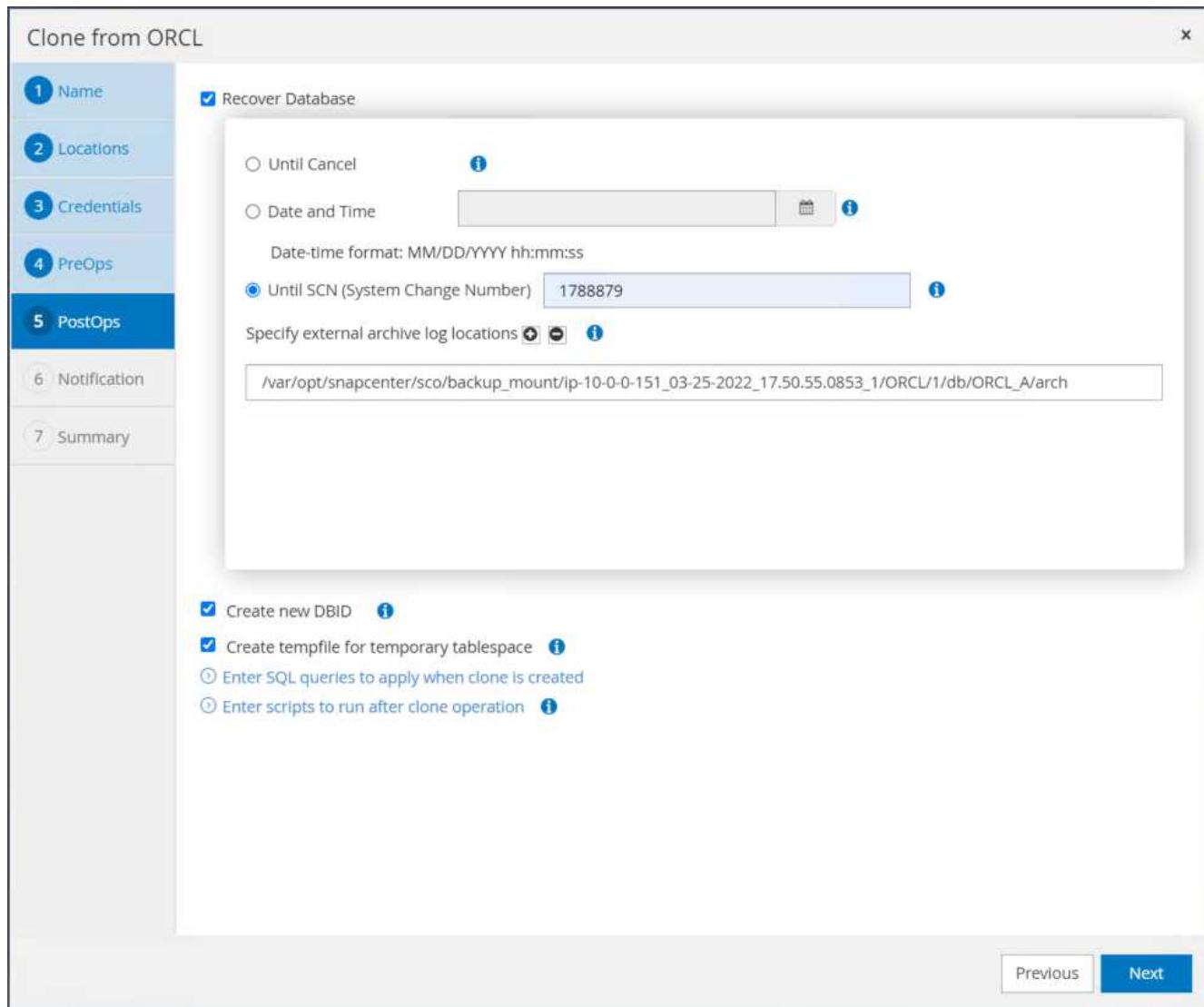
Oracle OS User

Oracle OS Group

Previous Next

The screenshot shows the 'Clone from ORCL' wizard at the 'Credentials' step (step 3). The left sidebar lists steps 1 through 7. The main area shows 'Database Credentials for the clone' with a dropdown for 'Credential name for sys user' set to 'None' and a port of '1521'. Below that, 'Oracle Home Settings' are configured with 'Oracle Home' as '/rdsdbbin/oracle', 'Oracle OS User' as 'rdsdb', and 'Oracle OS Group' as 'database'. At the bottom right are 'Previous' and 'Next' buttons.

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



9. Send the SMTP email settings if needed.

Clone from ORCL

Provide email settings i

Email preference	Never
From	From email
To	Email to
Subject	Notification

Attach job report

6 Notification

7 Summary

Previous **Next**

The screenshot shows a step-by-step cloning interface. The current step is 'Provide email settings'. It includes fields for email preference (set to 'Never'), sender ('From email'), recipient ('Email to'), subject ('Notification'), and an option to attach a job report. The 'Notification' tab is active. Navigation buttons 'Previous' and 'Next' are at the bottom right.

10. Clone the job summary, and click Finish to launch the clone job.

Clone from ORCL

	Summary
1 Name	Clone from backup
2 Locations	Clone SID
3 Credentials	Clone server
4 PreOps	Oracle home
5 PostOps	Oracle OS user
6 Notification	Oracle OS group
7 Summary	Datafile mountpaths
	Control files
	Redo groups
	Recovery scope
	Prescript full path
	Prescript arguments
	Postscript full path
	Postscript arguments
	Send email

Previous Finish

11. Validate the replica clone by reviewing the clone job log.

Job Details

Clone from backup 'ip-10-0-0-151_03-25-2022_17.50.55.0853_0'

- ✓ ▾ Clone from backup 'ip-10-0-0-151_03-25-2022_17.50.55.0853_0'
- ✓ ▾ ip-10-0-0-47.ec2.internal
 - ✓ ► Prescripts
 - ✓ ► Query Host Information
 - ✓ ► Prepare for Cloning
 - ✓ ► Cloning Resources
 - ✓ ► FileSystem Clone
 - ✓ ► Application Clone
 - ✓ ► Postscripts
 - ✓ ► Register Clone
 - ✓ ► Unmount Clone
 - ✓ ► Data Collection
 - ✓ ► Send EMS Messages

Task Name: ip-10-0-0-47.ec2.internal Start Time: 03/25/2022 9:08:32 PM End Time: 03/25/2022 9:12:03 PM

[View Logs](#) [Cancel Job](#) [Close](#)

The cloned database is registered in SnapCenter immediately.

NetApp SnapCenter®

Oracle Database

View	Database	Search databases	
Resources	ORCL	Single Instance	Host/Cluster: ip-10-0-0-151.ec2.internal Resource Group: orcl_full_bkup, orcl_log_bkup Policies: Oracle full backup, Oracle log backup Last Backup: 03/25/2022 9:10:09 PM Overall Status: Backup succeeded
	ORCLREAD	Single Instance	Host/Cluster: ip-10-0-0-47.ec2.internal Resource Group: (empty) Policies: (empty) Last Backup: (empty) Overall Status: Not protected

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.

[Oracle Database Migration from On-premises to FSx/EC2 via SnapMirror and BlueXP](#)

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.

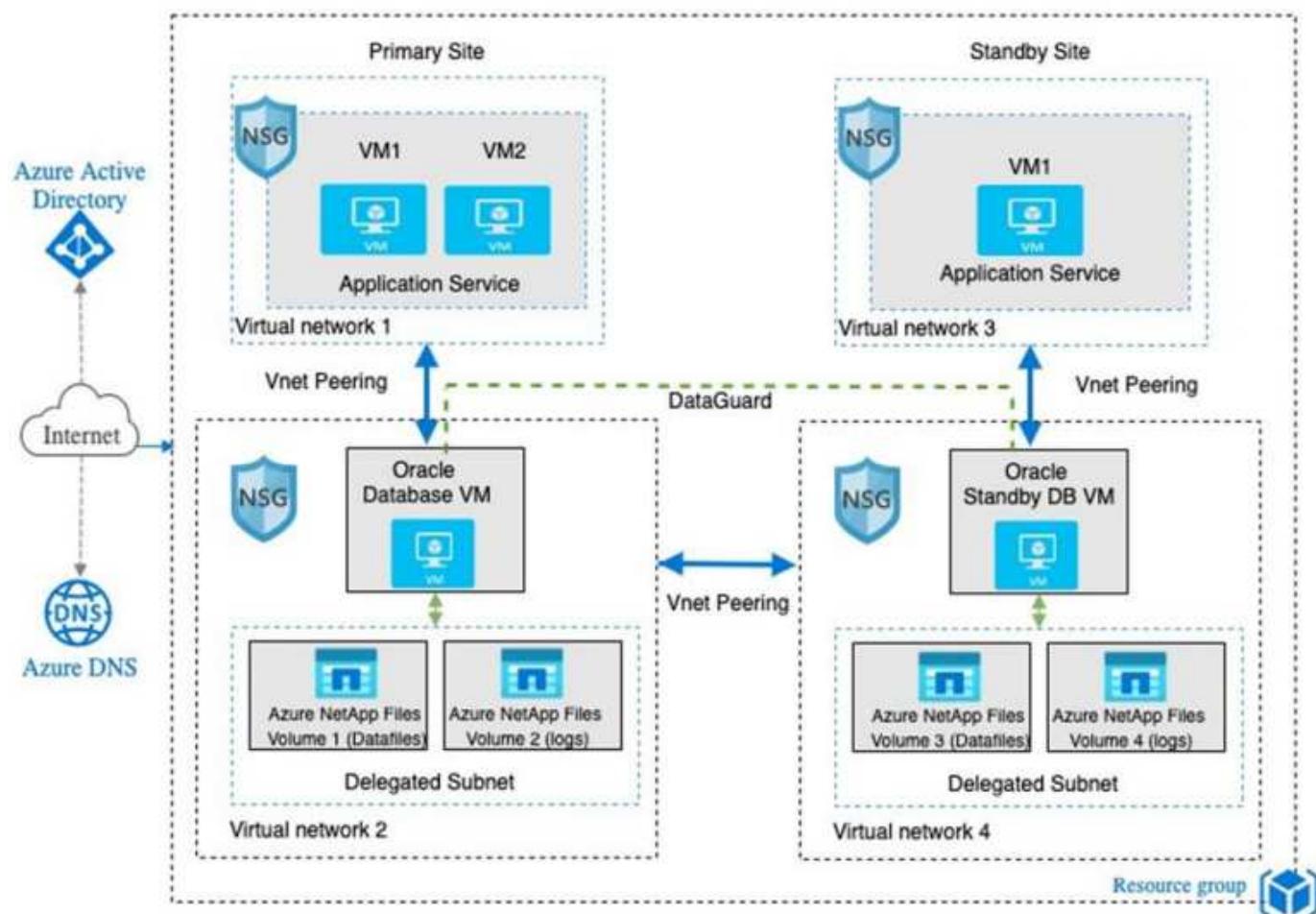
Solution Architecture

Previous: [Introduction](#).

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

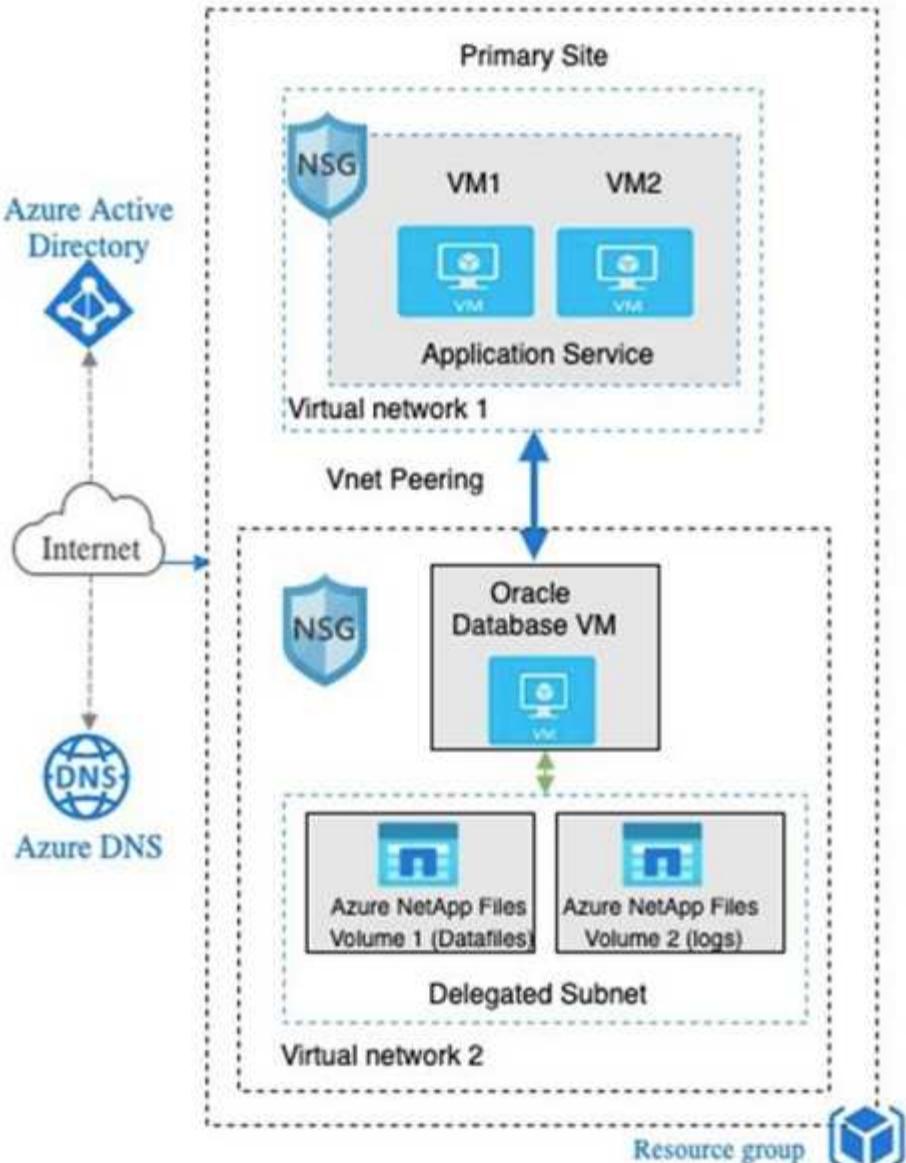
Within the environment, the Oracle compute instance is deployed via an Azure services VM console. There are multiple Azure instance types available from the console. NetApp recommends deploying a database-oriented Azure VM instance that meets your expected workload.

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

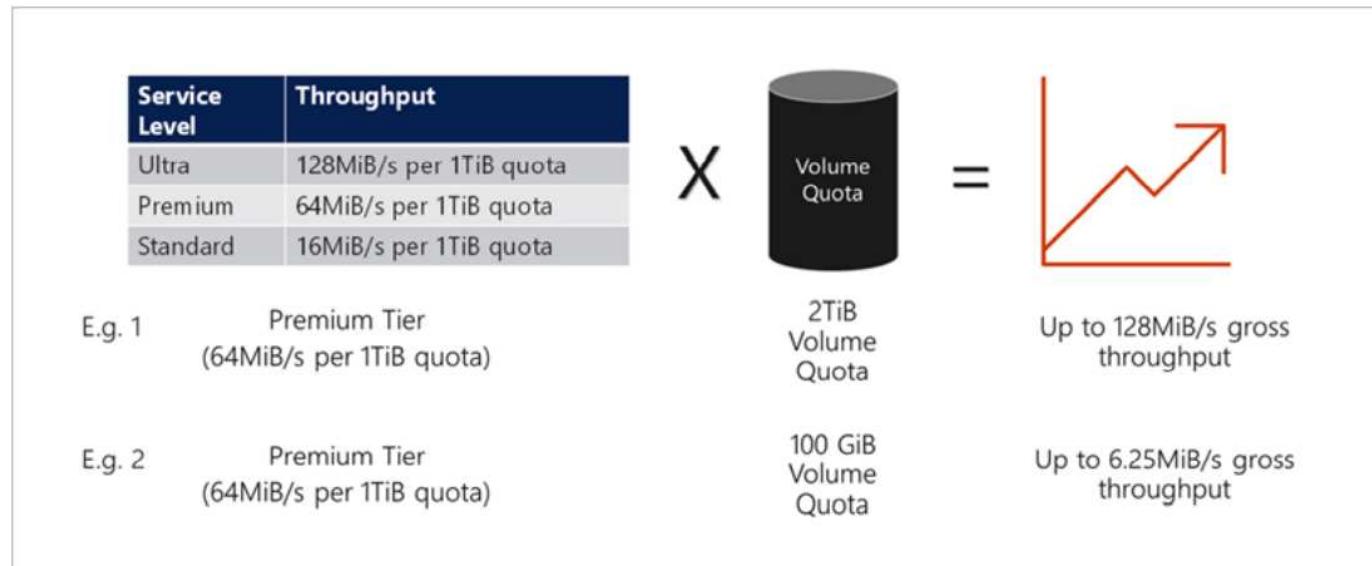
available throughput for all volumes in this capacity pool of 10x 64MBps, so 640MBps with 40,000 (16K) IOPs or 80,000 (8K) IOPs.

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 ($2\text{TiB} * 64\text{MiBps}$). This scenario applies regardless of the capacity pool size or the actual volume consumption.

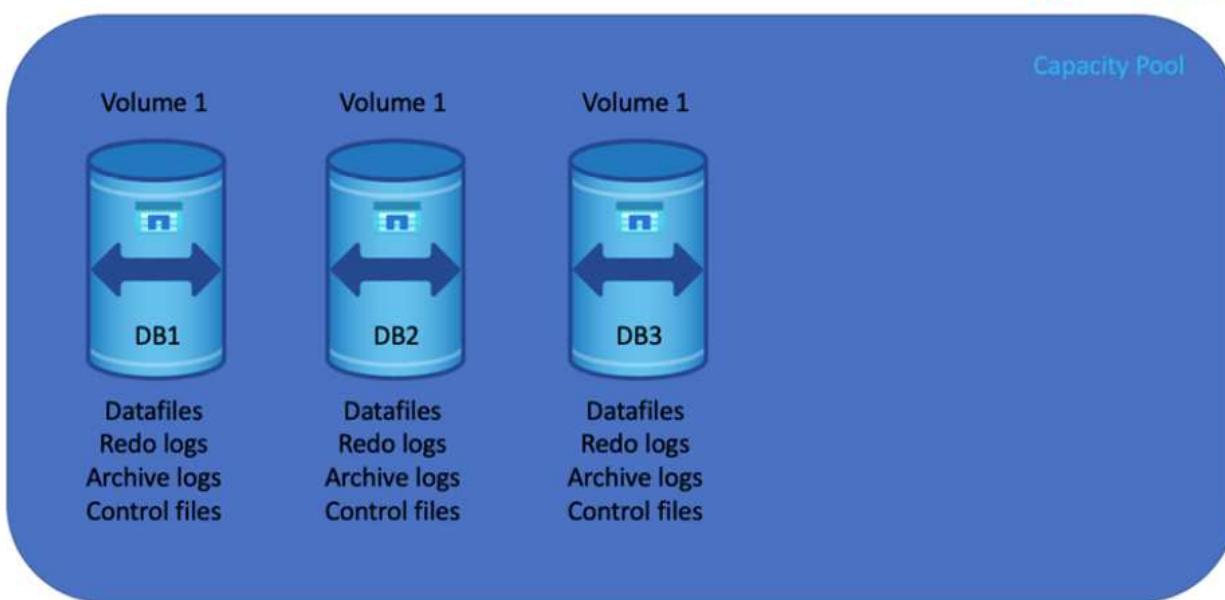
In example 2, a volume from a capacity pool with the Premium storage tier that is assigned 100GiB of quota is assigned a throughput limit of 6.25MiBps ($0.09765625\text{TiB} * 64\text{MiBps}$). This scenario applies regardless of the capacity pool size or the actual volume consumption.

Please note that the minimum volume size is 100GiB.

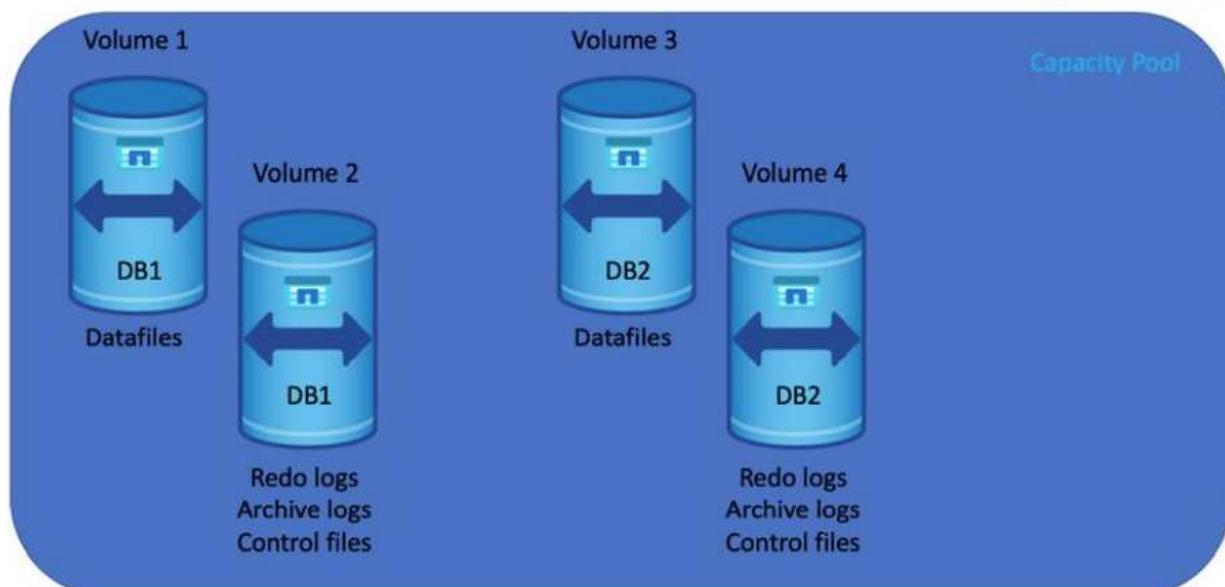
Storage layout and settings

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.



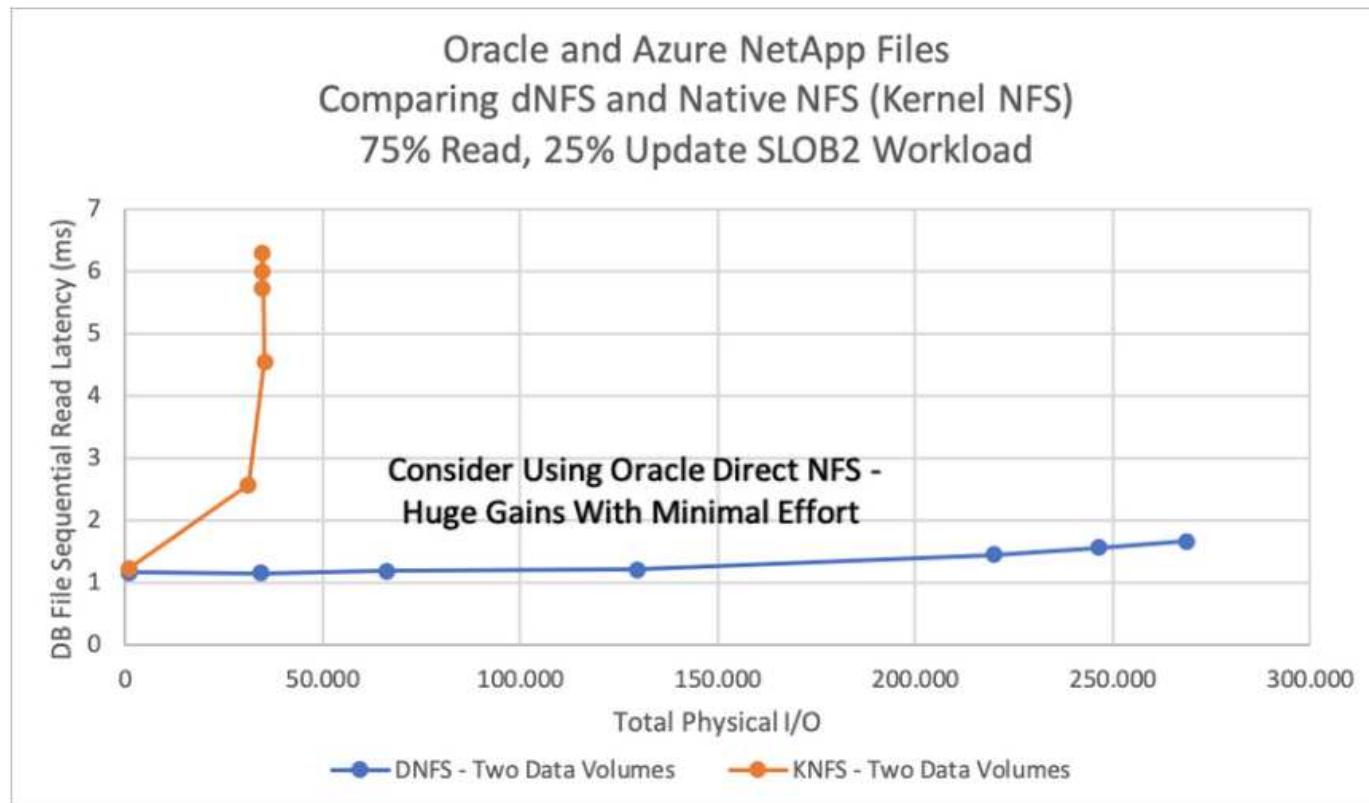
NFS configuration

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
<ul style="list-style-type: none"> • Control files • Data files • Redo logs 	<code>rw, bg, hard, vers=3, proto=tcp, timeo=600, rsize=65536, wsize=65536</code>
<ul style="list-style-type: none"> • ORACLE_HOME • ORACLE_BASE 	<code>rw, bg, hard, vers=3, proto=tcp, timeo=600, rsize=65536, wsize=65536</code>

 Before using dNFS, verify that the patches described in Oracle Doc 1495104.1 are installed. Starting with Oracle 12c, DNFS includes support for NFSv3, NFSv4, and NFSv4.1. NetApp support policies cover v3 and v4 for all clients, but, at the time of writing, NFSv4.1 is not supported for use with Oracle dNFS.

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.

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

Name	Type	Subscription	Resource group	Location	Status	Operating system	Size	Public IP address	Disks
acao-ora01	Virtual machine	Hybrid Cloud TME Onprem	TMEstorres	South Central US	Stopped (deallocated)	Linux	Standard_B4ms	13.65.63.157	1
ANFAVFW02JH	Virtual machine	Hybrid Cloud TME Onprem	ANFAVSQL2	West Europe	Running	Windows	Standard_DS2_v2	20.229.80.88	1
ANFAVSfio01	Virtual machine	Hybrid Cloud TME Onprem	anfavsqlrg	South Central US	Stopped (deallocated)	Linux	Standard_D32ds_v4	-	1
ANFAVSfioAZ1	Virtual machine	Hybrid Cloud TME Onprem	anfavsqlrg	South Central US	Running	Linux	Standard_E32as_v4	40.124.74.246	1
ANFAVSfioAZ2	Virtual machine	Hybrid Cloud TME Onprem	anfavsqlrg	South Central US	Stopped (deallocated)	Linux	Standard_E32as_v4	40.124.178.111	1
ANFAVSfioAZ3	Virtual machine	Hybrid Cloud TME Onprem	anfavsqlrg	South Central US	Stopped (deallocated)	Linux	Standard_E32as_v4	40.124.194.32	1
ANFAVSqlIDC	Virtual machine	Hybrid Cloud TME Onprem	anfavsqlrg	South Central US	Stopped (deallocated)	Windows	Standard_B4ms	-	1
ANFAVSqlIH	Virtual machine	Hybrid Cloud TME Onprem	anfavsqlrg	South Central US	Running	Windows	Standard_B2ms	70.37.66.218	1
ANFAVSqlIHZ	Virtual machine	Hybrid Cloud TME Onprem	anfavsqlrg	South Central US	Running	Windows	Standard_B2s	20.225.210.195	1
ANFCVOCM	Virtual machine	Hybrid Cloud TME Onprem	anfcval2	West Europe	Running	Linux	Standard_DS3_v2	-	1
ANFCVOORDC2	Virtual machine	Hybrid Cloud TME Onprem	anfcval2	West Europe	Running	Windows	Standard_B2s	-	1
ANFCVOORDemo	Virtual machine	Hybrid Cloud TME Onprem	anfcvordemorrg	West Europe	Running	Linux	Standard_E4s_v3	-	5
AVSCVOPerfinguest	Virtual machine	Hybrid Cloud TME Onprem	avscvoperfinguest-rg	West Europe	Stopped (deallocated)	Linux	Standard_DS15_v2	-	5

- 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.

[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 >](#)

[Home](#) > [Virtual machines](#) >

Create a virtual machine

Size * ⓘ

Standard_D8s_v3 - 8 vcpus, 32 GiB memory (\$273.02/month)

[See all sizes](#)

Administrator account

Authentication type ⓘ

 SSH public key Password

Username * ⓘ

azureuser



Password * ⓘ



Confirm password * ⓘ



Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * ⓘ

 None Allow selected ports

Select inbound ports *

SSH (22)



⚠ This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Licensing

If you have eligible Red Hat Enterprise Linux subscriptions that are enabled for Red Hat Cloud Access, you can use Azure Hybrid Benefit to attach your Red Hat subscriptions to this VM and save money on compute costs [Learn more ↗](#)

Your Azure subscription is currently not a part of Red Hat Cloud Access. In order to enable AHB for this VM, you must add this Azure subscription to Cloud Access. [Learn more ↗](#)

[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.

[Home](#) > [Virtual machines](#) >

Create a virtual machine

[Basics](#) [Disks](#) [Networking](#) [Management](#) [Advanced](#) [Tags](#) [Review + create](#)

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

Disk options

OS disk type * 

Premium SSD (locally-redundant storage) 

Delete with VM 



Enable encryption at host 



 Encryption at host is not registered for the selected subscription. [Learn more about enabling this feature](#)

Encryption type * 

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

Enable Ultra Disk compatibility 



Data disks for acao-ora01

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

LUN	Name	Size (GiB)	Disk type	Host caching	Delete with VM 
-----	------	------------	-----------	--------------	--

[Create and attach a new disk](#) [Attach an existing disk](#)

 Advanced

[Review + create](#)

[< Previous](#)

[Next : Networking >](#)

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

[Home](#) > [Virtual machines](#) >

Create a virtual machine

Network interface

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

Virtual network * ?

 ▼
[Create new](#)

Subnet * ?

 ▼
[Manage subnet configuration](#)

Public IP ?

 ▼
[Create new](#)

NIC network security group ?

- None
 Basic
 Advanced

Public inbound ports * ?

- None
 Allow selected ports

Select inbound ports *

 ▼

⚠ This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Delete public IP and NIC when VM is deleted ?

Enable accelerated networking ?

Load balancing

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

Place this virtual machine behind an existing load balancing solution?

[Review + create](#)

[< Previous](#)

[Next : Management >](#)

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

Home > Virtual machines >

Create a virtual machine

[Basics](#) [Disks](#) [Networking](#) [Management](#) [Advanced](#) [Tags](#) [Review + create](#)

Configure monitoring and management options for your VM.

Microsoft Defender for Cloud

Microsoft Defender for Cloud provides unified security management and advanced threat protection across hybrid cloud workloads. [Learn more ↗](#)

Your subscription is protected by Microsoft Defender for Cloud basic plan.

Monitoring

Boot diagnostics ⓘ

- Enable with managed storage account (recommended)
 Enable with custom storage account
 Disable

Enable OS guest diagnostics ⓘ

Identity

Enable system assigned managed identity ⓘ

Azure AD

Login with Azure AD ⓘ

RBAC role assignment of Virtual Machine Administrator Login or Virtual Machine User Login is required when using Azure AD login. [Learn more ↗](#)

Azure AD login now uses SSH certificate-based authentication. You will need to use an SSH client that supports OpenSSH certificates. You can use Azure CLI or Cloud Shell from the Azure Portal. [Learn more ↗](#)

Auto-shutdown

Enable auto-shutdown ⓘ

Backup

[Review + create](#)< PreviousNext : Advanced >

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.

[Home](#) > [Virtual machines](#) >

Create a virtual machine

[Basics](#) [Disks](#) [Networking](#) [Management](#) [Advanced](#) [Tags](#) [Review + create](#)

Add additional configuration, agents, scripts or applications via virtual machine extensions or cloud-init.

Extensions

Extensions provide post-deployment configuration and automation.

[Extensions](#) ⓘ[Select an extension to install](#)

VM applications

VM applications contain application files that are securely and reliably downloaded on your VM after deployment. In addition to the application files, an install and uninstall script are included in the application. You can easily add or remove applications on your VM after create. [Learn more ↗](#)

[Select a VM application to install](#)

Custom data

Pass a script, configuration file, or other data into the virtual machine **while it is being provisioned**. The data will be saved on the VM in a known location. [Learn more about custom data for VMs ↗](#)

[Custom data](#)

Your image must have a code to support consumption of custom data. If your image supports cloud-init, custom-data will be processed by cloud-init. [Learn more about custom data for VMs ↗](#)

User data

Pass a script, configuration file, or other data that will be accessible to your applications **throughout the lifetime of the virtual machine**. Don't use user data for storing your secrets or passwords. [Learn more about user data for VMs ↗](#)

[Enable user data](#)[Review + create](#)< PreviousNext : Tags >

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

[Home](#) > [Virtual machines](#) >

Create a virtual machine

Basics Disks Networking Management Advanced **Tags** Review + create

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

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

Name ⓘ	Value ⓘ	Resource
database	: oracle	12 selected <input type="button"/>
	:	12 selected <input type="button"/>

[Review + create](#)

[< Previous](#)

[Next : Review + create >](#)

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.

- 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.

The screenshot shows the Microsoft Azure portal's main dashboard. At the top, there's a blue header bar with the 'Microsoft Azure' logo and a search bar. Below the header, a row of service icons is displayed: 'Create a resource' (plus sign), 'Azure NetApp Files' (highlighted with a yellow box), 'Virtual networks', 'Virtual machines', 'Storage accounts', 'Users', 'Subscriptions', 'Azure Active Directory', 'Quickstart Center', and 'More services'. Underneath this, there's a section titled 'Resources' with tabs for 'Recent' and 'Favorite'. A table lists various resources with columns for 'Name', 'Type', and 'Last Viewed'. The resources listed include 'ANFAVSAcct' (NetApp account), 'ANFAVSAval' (Virtual network), 'acao-ora01' (Virtual machine), 'Hybrid Cloud TME Onprem' (Subscription), 'WEANFAVSAcct' (NetApp account), and several volumes and disks associated with it. At the bottom of the table, there's a link to 'See all'.

Name	Type	Last Viewed
ANFAVSAcct	NetApp account	a few seconds ago
ANFAVSAval	Virtual network	3 hours ago
acao-ora01	Virtual machine	5 days ago
Hybrid Cloud TME Onprem	Subscription	2 weeks ago
WEANFAVSAcct	NetApp account	2 weeks ago
ANFAVSAcct/CapPool/acao-ora01-u03	Volume	2 weeks ago
ANFAVSAcct/CapPool/acao-ora01-u02	Volume	2 weeks ago
ANFAVSAcct/CapPool/acao-ora01-u01	Volume	2 weeks ago
acao-ora01_OsDisk_1_673bad70ccce4709af8c1278e2bc97cb	Disk	2 weeks ago
acao-ora0166	Network Interface	3 weeks ago
TIMEtstres	Resource group	3 weeks ago

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

The screenshot shows the 'ANFAVSAcct' storage account page within the Azure NetApp Files service. The left sidebar has a tree view with nodes like 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Settings', 'Quota', 'Properties', 'Locks', 'Azure NetApp Files', 'Active Directory connections', 'Storage service', 'Capacity pools', 'Volumes' (highlighted with a yellow box), 'Data protection', 'Snapshot policies', 'Storage service add-ons', 'NetApp add-ons', 'Automation', 'Tasks (preview)', 'Export template', and 'Support + troubleshooting'. The main content area shows the 'Overview' tab with details such as Resource group (ANFAVSRG), Location (South Central US), Subscription (Hybrid Cloud TME Onprem), and Subscription ID (0efazdfb-917c-4497-b56a-b3f4eadb8111). It also displays tags (product_line : Field use - various). Below this, there's a section for 'Enterprise files storage, powered by NetApp' with links to 'Connect to Active Directory', 'Capacity pools', and 'Volumes'. At the bottom, there are buttons for 'View AD connections', 'View capacity pools', and 'View volumes'. The bottom left shows a page navigation bar with 'Page 1 of 1'.

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>**.

Microsoft Azure

Search resources, services, and docs (G+)

Home > Azure NetApp Files > ANFAVSAcct | Volumes >

ANFAVSAcct | Volumes

NetApp account

Create a volume

Basics Protocol Tags Review + create

Configure access to your volume.

Access

Protocol type: NFS SMB Dual-protocol

Configuration

File path *

Versions *

Kerberos: Enabled Disabled

LDAP: Enabled Disabled

Azure VMware Solution DataStore

Export policy

Configure the volume's export policy. This can be edited later. [Learn more](#)

Index	Allowed clients	Access	Root Access	...
1	172.30.137.142	Read & Write	On	...
2	172.30.137.142	Read & Write	On	...

Review + create < Previous Next : Tags >

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

Microsoft Azure

Search resources, services, and docs (G+)

Home > Azure NetApp Files > ANFAVSAcct | Volumes >

ANFAVSAcct | Volumes

NetApp account

Search (Ctrl+ /) Add volume ...

Overview

Activity log

Access control (IAM)

Tags

Settings

Quota

Properties

Locks

Azure NetApp Files

Active Directory connections

Storage service

Capacity pools

Volumes

Snapshot policies

Storage service add-ons

NetApp add-ons

Automation

Tasks (preview)

Export template

Support + troubleshooting

New Support Request

+ Add volume ...

Search volumes

Name	↑↓	Quota
anf2-z1-stdds01	200 GiB	
anf2-z1-stdds02	200 GiB	
anf2-z1-stdds03	100 GiB	
anf2-z1-stdds04	100 GiB	
anf2-z1-stdds05	100 GiB	
anf2-z1-stdds06	100 GiB	
anf2-z1-stdds07	100 GiB	
anf2-z1-stdds08	100 GiB	
anf-z1-stdds01	6 TiB	
anf-z1-stdds02	200 GiB	
anf-z1-stdds03	1 TiB	
anf-z1-stdds04	200 GiB	
anf-z1-stdds06	200 GiB	
anf-z1-stdds07	200 GiB	
anf-z1-stdds08	200 GiB	
anf-zq-stdds05	1 TiB	
vol1	1 TiB	
vol3basic	100 GiB	
volnfsbasic	100 GiB	
volnfsstd	100 GiB	
volnfsstdnew	100 GiB	
zone1basic	6 TiB	
zone2basic	100 GiB	

Create a volume ...

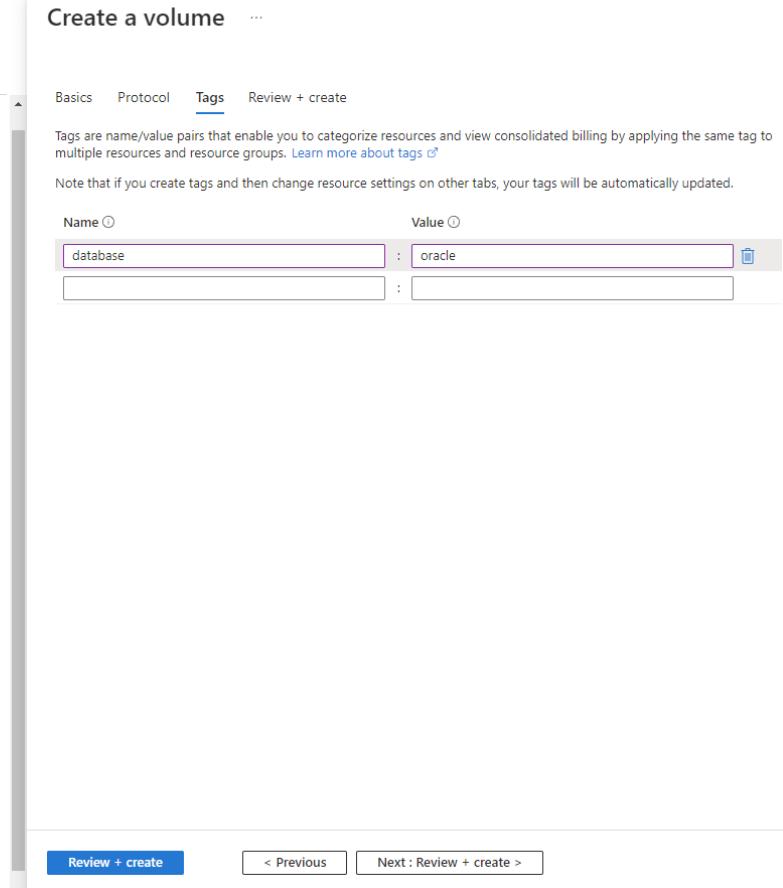
Basics Protocol Tags Review + create

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

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

Name Value database : oracle

Review + create < Previous Next : Review + create >



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

Create a volume

Validation passed

Basics

Subscription	Hybrid Cloud TME Onprem
Resource group	ANFAVSRG
Region	South Central US
Volume name	acao-ora01-u01
Capacity pool	CapPool
Service level	Ultra
Quota	100 GiB
Encryption key source	Microsoft.NetApp
Availability Zone	None

Networking

Virtual network	ANFASVal (172.30.136.64/26,172.30.137.128/25,172.30.152.0/27)
Delegated subnet	ANF_Sub (172.30.136.64/26)
Network features	Standard

Protocol

Protocol	NFSv3
File path	acao-ora01-u01

Tags

database	oracle
----------	--------

Create < Previous Next > Download a template for automation

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'.

```

- 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
```

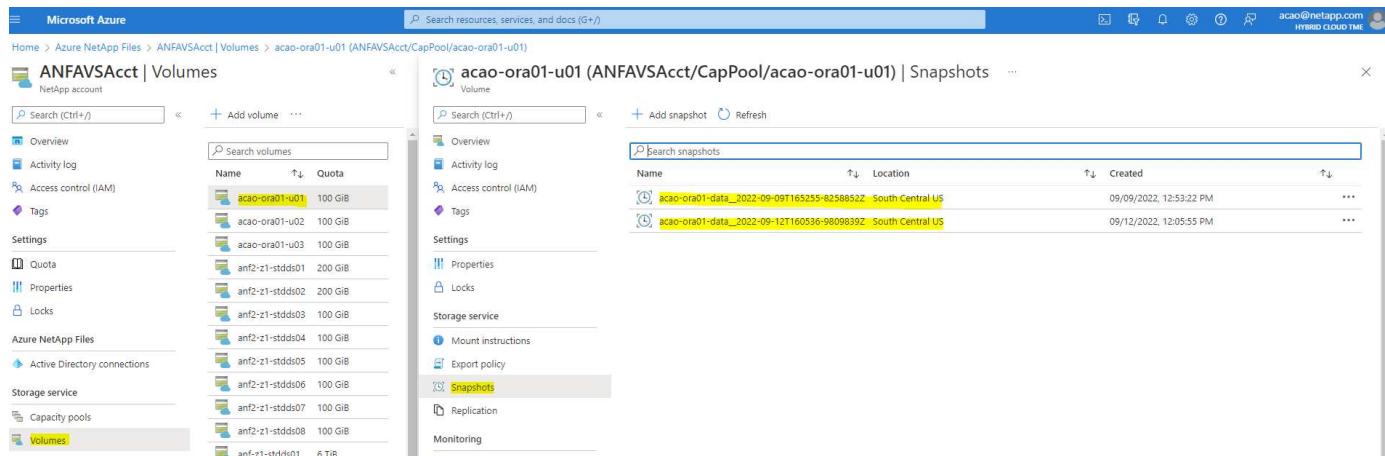
Crontab schedule:

```
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.



Name	Location	Created
acao-ora01-data_2022-09-09T165255-82588522	South Central US	09/09/2022, 12:53:22 PM
acao-ora01-data_2022-09-12T160536-98098392	South Central US	09/12/2022, 12:05:55 PM

The screenshot shows the Microsoft Azure portal interface for NetApp Files. On the left, the 'Volumes' section is selected, displaying a list of volumes including 'acao-ora01-u01' (100 GiB), 'acao-ora01-u02' (100 GiB), 'acao-ora01-u03' (100 GiB), 'anf2-z1-stdds01' (200 GiB), 'anf2-z1-stdds02' (200 GiB), 'anf2-z1-stdds03' (100 GiB), 'anf2-z1-stdds04' (100 GiB), 'anf2-z1-stdds05' (100 GiB), 'anf2-z1-stdds06' (100 GiB), 'anf2-z1-stdds07' (100 GiB), and 'anf2-z1-stdds08' (100 GiB). The 'acao-ora01-u03' volume is highlighted.

On the right, a detailed view of the 'acao-ora01-u03' volume is shown, specifically the 'Schemas' tab. It lists several log snapshots, each with a timestamp and location:

Name	Location	Created
acao-ora01-data_2022-09-12T160628-87547982	South Central US	09/12/2022, 12:06:31 PM
acao-ora01-log_2022-09-12T164501-79659252	South Central US	09/12/2022, 12:45:04 PM
acao-ora01-log_2022-09-12T171501-80326612	South Central US	09/12/2022, 01:15:04 PM
acao-ora01-log_2022-09-12T173001-47879192	South Central US	09/12/2022, 01:30:04 PM
acao-ora01-log_2022-09-12T174501-50106142	South Central US	09/12/2022, 01:45:04 PM
acao-ora01-log_2022-09-12T180053-50298742	South Central US	09/12/2022, 02:00:55 PM
acao-ora01-log_2022-09-12T181502-38570272	South Central US	09/12/2022, 02:15:05 PM
acao-ora01-log_2022-09-12T183002-44078952	South Central US	09/12/2022, 02:30:07 PM

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
```

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

The screenshot shows the Microsoft Azure Azure NetApp Files dashboard. On the left, there's a sidebar with options like 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Settings', 'Quota', 'Properties', 'Locks', 'Azure NetApp Files', 'Storage service', 'Capacity pools', 'Volume' (which is selected), and 'Application volume groups'. The main area shows the 'ANFAVSAcc1' account with 'Volumes'. One volume, 'acao-ora01-u03', is highlighted. To the right, under 'Solutions', there's a table for 'Solutions' and a 'Revert volume' button, which is also highlighted with a yellow box.

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

Revert volume to snapshot

Revert volume acao-ora01-u03 to snapshot acao-ora01-log_2022-09-13T134502-04499192

acao-ora01-u03 (ANFAVSAcc/CapPool/acao-ora01-u03) | Snapshots

Name	Location
acao-ora01-data_2022-09-12T160628-87547982	South Central US
acao-ora01-log_2022-09-13T120122-61736452	South Central US
acao-ora01-log_2022-09-13T121501-67804172	South Central US
acao-ora01-log_2022-09-13T123001-80005482	South Central US
acao-ora01-log_2022-09-13T124501-74729672	South Central US
acao-ora01-log_2022-09-13T131501-62704482	South Central US
acao-ora01-log_2022-09-13T133001-77736192	South Central US
acao-ora01-log_2022-09-13T134502-04499192	South Central US

acao-ora01-u03 (ANFAVSAcc/CapPool/acao-ora01-u03) | Volumes

Name	Quota
acao-ora01-u01	100 GiB
acao-ora01-u02	100 GiB
acao-ora01-u03	100 GiB
anf2-z1-std5d01	200 GiB
anf2-z1-std5d02	200 GiB
anf2-z1-std5d03	100 GiB
anf2-z1-std5d04	100 GiB
anf2-z1-std5d05	100 GiB
anf2-z1-std5d06	100 GiB
anf2-z1-std5d07	100 GiB
anf2-z1-std5d08	100 GiB
anf2-z1-std5d01	6 TiB
anf2-z1-std5d02	200 GiB
anf2-z1-std5d03	1 TiB
anf2-z1-std5d04	200 GiB
anf2-z1-std5d06	200 GiB
anf2-z1-std5d07	200 GiB
anf2-z1-std5d08	200 GiB
anf-zq-std5d05	1 TiB
vol1	1 TiB
vol3basic	100 GiB

Revert **Cancel**

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

Revert volume to snapshot

Revert volume acao-ora01-u02 to snapshot acao-ora01-data_2022-09-13T100002-43474562

acao-ora01-u02 (ANFAVSAcc/CapPool/acao-ora01-u02) | Snapshots

Name	Location	Created
acao-ora01-data_2022-09-13T020001-95033842	South Central US	09/12/2022, 10:00:19 PM
acao-ora01-data_2022-09-13T040001-93413402	South Central US	09/13/2022, 12:00:18 AM
acao-ora01-data_2022-09-13T060002-12409142	South Central US	09/13/2022, 02:00:19 AM
acao-ora01-data_2022-09-13T080001-83834982	South Central US	09/13/2022, 04:00:18 AM
acao-ora01-data_2022-09-13T100002-43474562	South Central US	09/13/2022, 06:00:18 AM
acao-ora01-data_2022-09-13T120002-34062902	South Central US	09/13/2022, 08:00:18 AM
acao-ora01-data_2022-09-13T140001-85298172	South Central US	09/13/2022, 10:00:17 AM

acao-ora01-u02 (ANFAVSAcc/CapPool/acao-ora01-u02) | Volumes

Name	Quota
acao-ora01-u01	100 GiB
acao-ora01-u02	100 GiB
acao-ora01-u03	100 GiB
anf2-z1-std5d01	200 GiB
anf2-z1-std5d02	200 GiB
anf2-z1-std5d03	100 GiB
anf2-z1-std5d04	100 GiB
anf2-z1-std5d05	100 GiB
anf2-z1-std5d06	100 GiB
anf2-z1-std5d07	100 GiB
anf2-z1-std5d08	100 GiB
anf2-z1-std5d01	6 TiB
anf2-z1-std5d02	200 GiB

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__29f21gb5_.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__29f21gb5_.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_inViolations 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](#)
- [Part 2: Variables and Running the Playbook](#)

CLI Deployment

- [Part 1: Getting Started, Requirements, Automation Details and Ansible Control Host Setup](#)
- [Part 2: Variables and Running the Playbook](#)

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 it 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 <ul style="list-style-type: none">- netapp-lib- xmltodict- 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

```

- d. Paste the following content into Injector Configuration:

```

extra_vars:
  password: '{{ password }}'
  username: '{{ username }}'
  vsadmin_password: '{{ vsadmin_password }}'

```

3. 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

```
<style>
div {
position: relative;
}
div button {
position: absolute;
top: 0;
right: 0;
}
button {
transition-duration: 0.4s;
background-color: white;
color: #1563a3;
border: 2px solid #1563a3;
}
button:hover {
background-color: #1563a3;
color: white;
}
#more_datastores_nfs {
display: block;
}
#more_datastores_nfs_button {
display: none;
}
</style>
<div class="listingblock"><div class="content"><div><button id="copy-
button1" onclick="CopyClassText1()">Copy</button></div><pre><code><div
class="CopyMeClass1" id="CopyMeID1">
#####
#####          Host Variables Configuration          #####
#####
# Add your Oracle Host
ansible_host: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
```

```

decoration:underline;"/><i>10.61.180.15</i></span>

# Oracle db log archive mode: true - ARCHIVELOG or false - NOARCHIVELOG
log_archive_mode: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;"><i>true</i></span>

# 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: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;"><i>cdb2</i></span>
pdb_num: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;"><i>3</i></span>
pdb_name: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;"><i>"{{ oracle_sid }}_pdb"</i></span>

# CDB listener port, use different listener port for additional CDB on same host
listener_port: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;"><i>1523</i></span>

# 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: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;"><i>5464</i></span>

# 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: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;"><i>DBEXPRESS</i></span>
em_express_port: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;"><i>5501</i></span>

# "{{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: ""<span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>{{groups.oracle[0]}}_u01</i></span>", aggr_name: <span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>aggr01_node01</i></span>, lif: <span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>172.21.94.200</i></span>, size: <span <div contenteditable='true'><i>25</i></span>}"
    - {vol_name: ""<span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>{{groups.oracle[0]}}_u02</i></span>", aggr_name: <span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>aggr01_node01</i></span>, lif: <span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>172.21.94.200</i></span>, size: <span <div contenteditable='true'><i>25</i></span>}"
      - {vol_name: ""<span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>{{groups.oracle[0]}}_u03</i></span>", aggr_name: <span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>aggr01_node01</i></span>, lif: <span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>172.21.94.200</i></span>, size: <span <div contenteditable='true'><i>25</i></span>}"
        <a id="more_datastores_nfs" href="javascript:datastoredropdown();">More NFS volumes</a><div id="select_more_datastores_nfs"></div><a id="more_datastores_nfs_button"
          href="javascript:adddatastorevolumes();">Enter NFS volumes' details</a><div id="extra_datastores_nfs"></div>
        </div></code></pre></div></div>
<script>
function CopyClassText1(){
  var textToCopy = document.getElementById("CopyMeID1");

```

```

var currentRange;
if(document.getSelection().rangeCount > 0)
{
    currentRange = document.getSelection().getRangeAt(0);
    window.getSelection().removeRange(currentRange);
}
else
{
    currentRange = false;
}
var CopyRange = document.createRange();
CopyRange.selectNode(textToCopy);
window.getSelection().addRange(CopyRange);
document.getElementById("more_datastores_nfs").style.display = "none";
var command = document.execCommand("copy");
if (command)
{
    document.getElementById("copy-button1").innerHTML = "Copied!";
    setTimeout(revert_copy, 3000);
}
window.getSelection().removeRange(CopyRange);
if(currentRange)
{
    window.getSelection().addRange(currentRange);
}
}

function revert_copy() {
    document.getElementById("copy-button1").innerHTML = "Copy";
    document.getElementById("more_datastores_nfs").style.display =
"block";
}

function datastoredropdown() {
    document.getElementById("more_datastores_nfs").style.display = "none";
    document.getElementById("more_datastores_nfs_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper = document.getElementById("select_more_datastores_nfs");
    while (x < 100) {
        buildup += '<option value="' + x + '">' + x + '</option>';
        x++;
    }
    myHTML += '<a id="more_datastores_nfs">How many extra NFS volumes do
you wish to add?</a><select name="number_of_extra_datastores_nfs"

```

```

id="number_of_extra_datastores_nfs">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function adddatastorevolumes() {
    var y =
document.getElementById("number_of_extra_datastores_nfs").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_datastores_nfs");
    while (j < y) {
        j++;
        myHTML += ' - {vol_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"><i>{{groups.oracle[0]}}_u01</i></span>,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>aggr01_node02</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"><i>172.21.94.201</i></span>,
size: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>25</i></span>}<br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_datastores_nfs").style.display =
"none";
    document.getElementById("more_datastores_nfs_button").style.display =
"none";
}

</script>

```

- Fill in all variables in the blue fields.
- After completing variables input, click the Copy button on the form to copy all variables to be transferred to AWX or Tower.
- Navigate back to AWX or Tower and go to Resources → Hosts, and select and open the Oracle server configuration page.
- Under the Details tab, click edit and paste the copied variables from step 1 to the Variables field under the YAML tab.
- Click Save.
- 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.

VARS

```
<style>
div {
position: relative;
}
div button {
position: absolute;
top: 0;
right: 0;
}
button {
transition-duration: 0.4s;
background-color: white;
color: #1563a3;
border: 2px solid #1563a3;
}
button:hover {
background-color: #1563a3;
color: white;
}
#more_storage_vlans {
display: block;
}
#more_storage_vlans_button {
display: none;
}
#more_nfs_volumes {
display: block;
}
#more_nfs_volumes_button {
display: none;
}
</style>
<div class="listingblock"><div class="content"><div><button id="copy-
button" onclick="CopyClassText()">Copy</button></div><pre><code><div
class="CopyMeClass" id="CopyMeID">
#####
##### 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: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>ontap</i></span>

#CA_signed_certificates (ONLY CHANGE to 'true' IF YOU ARE USING CA SIGNED
CERTIFICATES)
ca_signed_certs: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"><i>false</i></span>

#Names of the Nodes in the ONTAP Cluster
nodes:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"><i>AFF-
01</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"><i>AFF-
02</i></span>

#Storage VLANs
#Add additional rows for vlans as necessary
storage_vlans:
  - {vlan_id: ""<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"><i>203</i></span>", name: ""<span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"><i>infra_NFS</i></span>",
protocol: ""<span <div
contenteditable="true"/><i>NFS</i></span>",
More Storage VLANs</a><div id="select_more_storage_vlans"></div><a
id="more_storage_vlans_button" href="javascript:addstoragevlans();">Enter
Storage VLANs details</a><div id="extra_storage_vlans"></div>
#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: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"><i>aggr01_node01</i></span>}
  - {aggr_name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"><i>aggr01_node02</i></span>}

#SVM name
svm_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>ora_svm</i></span>

# SVM Management LIF Details
svm_mgmt_details:
  - {address: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"><i>172.21.91.100</i></span>, netmask: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"><i>255.255.255.0</i></span>,
home_port: <span <div contenteditable="true"/><i>e0M</i></span>}

# 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: "<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"><i>{{groups.oracle[0]}}_u01</i></span>",
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>aggr01_node01</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-

```

```

style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span>,
size: <span <div contenteditable="true"/><i>25</i></span>
    - {vol_name: &quot<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"><i>{{groups.oracle[0]}}_u02</i></span>&quot,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>aggr01_node01</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"><i>172.21.94.200</i></span>,
size: <span <div contenteditable="true"/><i>25</i></span>
    - {vol_name: &quot<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"><i>{{groups.oracle[0]}}_u03</i></span>&quot,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>aggr01_node01</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"><i>172.21.94.200</i></span>,
size: <span <div contenteditable="true"/><i>25</i></span>
<a id="more_nfs_volumes" href="javascript:nfsvolumesdropdown();">More NFS
volumes</a><div id="select_more_nfs_volumes"></div><a
id="more_nfs_volumes_button" href="javascript:adnnfsvolumes();">Enter NFS
volumes' details</a><div id="extra_nfs_volumes"></div>

#NFS LIFs IP address and netmask
nfs_lifs_details:
    - address: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>172.21.94.200</i></span> #for node-1
        netmask: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>255.255.255.0</i></span>
    - address: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>172.21.94.201</i></span> #for node-2
        netmask: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>255.255.255.0</i></span>

#NFS client match
client_match: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"><i>172.21.94.0/24</i></span>

#####

```

```

### Linux env specific config variables ###

#####
#NFS Mount points for Oracle DB volumes
mount_points:
  - <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>/u01</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>/u02</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>/u03</i></span>

# 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: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>1234</i></span>

# RedHat subscription username and password
redhat_sub_username: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>xxx</i></span>
redhat_sub_password: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>xxx</i></span>

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

db_domain: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>your.domain.com</i></span>

# Set initial password for all required Oracle passwords. Change them after installation.
initial_pwd_all: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>netapp123</i></span>

</div></code></pre></div></div>

```

```

<script>
function CopyClassText() {
    var textToCopy = document.getElementById("CopyMeID");
    var currentRange;
    if(document.getSelection().rangeCount > 0)
    {
        currentRange = document.getSelection().getRangeAt(0);
        window.getSelection().removeRange(currentRange);
    }
    else
    {
        currentRange = false;
    }
    var CopyRange = document.createRange();
    CopyRange.selectNode(textToCopy);
    window.getSelection().addRange(CopyRange);
    document.getElementById("more_storage_vlans").style.display = "none";
    document.getElementById("more_nfs_volumes").style.display = "none";
    var command = document.execCommand("copy");
    if (command)
    {
        document.getElementById("copy-button").innerHTML = "Copied!";
        setTimeout(revert_copy, 3000);
    }
    window.getSelection().removeRange(CopyRange);
    if(currentRange)
    {
        window.getSelection().addRange(currentRange);
    }
}
function revert_copy() {
    document.getElementById("copy-button").innerHTML = "Copy";
    document.getElementById("more_storage_vlans").style.display =
"block";
    document.getElementById("more_nfs_volumes").style.display = "block";
}
function storagevlandropdown() {
    document.getElementById("more_storage_vlans").style.display = "none";
    document.getElementById("more_storage_vlans_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper = document.getElementById("select_more_storage_vlans");
    while (x < 10) {
        buildup += '<option value="' + x + '">' + x + '</option>';

```

```

        x++;
    }
    myHTML += '<a id="more_storage_vlans_info">How many extra VLANs do you
    wish to add?</a><select name="number_of_extra_storage_vlans"
    id="number_of_extra_storage_vlans">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function addstoragevlans() {
    var y =
document.getElementById("number_of_extra_storage_vlans").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_storage_vlans");
    while (j < y) {
        j++;
        myHTML += ' - {vlan_id: " + j + ", name: " + wrapper.innerHTML + ", protocol: " + wrapper.protocol + ", port: " + wrapper.port + ", ip: " + wrapper.ip + ", subnet: " + wrapper.subnet + ", gateway: " + wrapper.gateway + ", broadcast: " + wrapper.broadcast + ", mac: " + wrapper.mac + ", type: " + wrapper.type + ", status: " + wrapper.status + ", description: " + wrapper.description + "}";
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_storage_vlans").style.display =
"none";
    document.getElementById("more_storage_vlans_button").style.display =
"none";
}
function nfsvolumesdropdown() {
    document.getElementById("more_nfs_volumes").style.display = "none";
    document.getElementById("more_nfs_volumes_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper = document.getElementById("select_more_nfs_volumes");
    while (x < 100) {
        buildup += '<option value=" + x + ">' + x + '</option>';
        x++;
    }
    myHTML += '<a id="more_nfs_volumes_info">How many extra NFS volumes do
    you wish to add?</a><select name="number_of_extra_nfs_volumes"
    id="number_of_extra_nfs_volumes">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}

```

```

}

function addnfsvolumes() {
    var y = document.getElementById("number_of_extra_nfs_volumes").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_nfs_volumes");
    while (j < y) {
        j++;
        myHTML += ' - {vol_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"><i>rtpora04_u01</i></span>, aggr_name: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"><i>aggr01_node02</i></span>,
lif: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>172.21.94.201</i></span>, size: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"><i>25</i></span>}<br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_nfs_volumes").style.display =
"none";
    document.getElementById("more_nfs_volumes_button").style.display =
"none";
}

</script>

```

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.

 - d. Click Next and then Launch to start the job.
 - e. Click View → Jobs to monitor the job output and progress.
 - f. 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.
 - g. Click Next and then Launch to start the job.
 - h. Click View → Jobs to monitor the job output and progress
 - i. After the ontap_config role has completed, run the process again for linux_config.
 - j. Navigate to Resources → Templates.
 - k. Select the desired template and then click Launch.
 - l. 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.
 - m. Click Next and then Launch to start the job.
 - n. Select View → Jobs to monitor the job output and progress.
 - o. After the linux_config role has completed, run the process again for oracle_config.
 - p. Go to Resources → Templates.
 - q. Select the desired template and then click Launch.
 - r. 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.
 - s. Click Next and then Launch to start the job.
 - t. 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.

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.

```
sqlplus / as sysdba
```

```
[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>
```

```
select name, log_mode from v$database;
```

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

```
NAME LOG_MODE
```

```
-----
```

```
CDB2 ARCHIVELOG
```

```
show pdbs;
```

```
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 the Oracle listener configuration with the following command. Change to the appropriate listener port and database service name.

```
sqlplus system@//localhost:1523/cdb2_pdb1.cie.netapp.com
```

[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

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<div>
[ontap]
<div>
<span <div contenteditable="false" style="color:#7EAF97
; font-weight:bold; font-style:italic; text-
decoration:;"/>10.61.184.183<i></i></span>
</div>
#Oracle hosts<div>
<div>
[oracle]<div>
<span <div contenteditable="false" style="color:#7EAF97
; font-weight:bold; font-style:italic; text-
decoration:;"/>rtpora01<i></i></span>
<div>
<span <div contenteditable="false" style="color:#7EAF97
; font-weight:bold; font-style:italic; text-
decoration:;"/>rtpora02<i></i></span>
</div>
```

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

```
<style>
div {
position: relative;
}
div button {
position: absolute;
top: 0;
right: 0;
}
button {
```

```

transition-duration: 0.4s;
background-color: white;
color: #1563a3;
border: 2px solid #1563a3;
}
button:hover {
background-color: #1563a3;
color: white;
}
#more_datastores_nfs {
display: block;
}
#more_datastores_nfs_button {
display: none;
}
</style>
<div class="listingblock"><div class="content"><div><button id="copy-
button1" onclick="CopyClassText1()">Copy</button></div><pre><code><div
class="CopyMeClass1" id="CopyMeID1">
#####
#####          Host Variables Configuration          #####
#####
# Add your Oracle Host
ansible_host: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"><i>10.61.180.15</i></span>

# Oracle db log archive mode: true - ARCHIVELOG or false - NOARCHIVELOG
log_archive_mode: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"><i>true</i></span>

# 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: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"><i>cdb2</i></span>
pdb_num: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"><i>3</i></span>
pdb_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"><i>"{{ oracle_sid }}_pdb"</i></span>

```

```

# CDB listener port, use different listener port for additional CDB on
same host
listener_port: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"><i>1523</i></span>

# 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: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"><i>5464</i></span>

# 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: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"><i>DBEXPRESS</i></span>
em_express_port: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"><i>5501</i></span>

# "{{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: ""<span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>{{groups.oracle[0]}}_u01</i></span>",
    aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>aggr01_node01</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"><i>172.21.94.200</i></span>,
    size: <span <div contenteditable="true"/><i>25</i></span>}
  - {vol_name: ""<span <div contenteditable="true"

```

```

style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>{{groups.oracle[0]}}_u02</i></span>" ,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>aggr01_node01</i></span>, lif: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span>,
size: <span <div contenteditable="true"/><i>25</i></span>
- {vol_name: ""<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>{{groups.oracle[0]}}_u03</i></span>" ,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>aggr01_node01</i></span>, lif: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>172.21.94.200</i></span>,
size: <span <div contenteditable="true"/><i>25</i></span>}
<a id="more_datastores_nfs" href="javascript:datastoredropdown();">More NFS volumes</a><div id="select_more_datastores_nfs"></div><a id="more_datastores_nfs_button"
href="javascript:adddatastorevolumes();">Enter NFS volumes' details</a><div id="extra_datastores_nfs"></div>
</div></code></pre></div>
<script>
function CopyClassText1() {
    var textToCopy = document.getElementById("CopyMeID1");
    var currentRange;
    if(document.getSelection().rangeCount > 0)
    {
        currentRange = document.getSelection().getRangeAt(0);
        window.getSelection().removeRange(currentRange);
    }
    else
    {
        currentRange = false;
    }
    var CopyRange = document.createRange();
    CopyRange.selectNode(textToCopy);
    window.getSelection().addRange(CopyRange);
    document.getElementById("more_datastores_nfs").style.display = "none";
    var command = document.execCommand("copy");
    if (command)
    {
        document.getElementById("copy-button1").innerHTML = "Copied!";
        setTimeout(revert_copy, 3000);
    }
}

```

```

        window.getSelection().removeRange(CopyRange);
        if(currentRange)
        {
            window.getSelection().addRange(currentRange);
        }
    }

    function revert_copy() {
        document.getElementById("copy-button1").innerHTML = "Copy";
        document.getElementById("more_datastores_nfs").style.display =
"block";
    }

    function datastoredropdown() {
        document.getElementById("more_datastores_nfs").style.display = "none";
        document.getElementById("more_datastores_nfs_button").style.display =
"block";
        var x=1;
        var myHTML = '';
        var buildup = '';
        var wrapper = document.getElementById("select_more_datastores_nfs");
        while (x < 100) {
            buildup += '<option value="' + x + '">' + x + '</option>';
            x++;
        }
        myHTML += '<a id="more_datastores_nfs">How many extra NFS volumes do
you wish to add?</a><select name="number_of_extra_datastores_nfs"
id="number_of_extra_datastores_nfs">' + buildup + '</select>';
        wrapper.innerHTML = myHTML;
    }

    function adddatastorevolumes() {
        var y =
document.getElementById("number_of_extra_datastores_nfs").value;
        var j=0;
        var myHTML = '';
        var wrapper = document.getElementById("extra_datastores_nfs");
        while (j < y) {
            j++;
            myHTML += ' - {vol_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"><i>{{groups.oracle[0]}}_u01</i></span>,
aggr_name: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>aggr01_node02</i></span>, lif: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"><i>172.21.94.201</i></span>,
size: <span <div contenteditable="true" style="color:#004EFF; font-

```

```

        weight:bold; font-style:italic; text-
decoration:underline;"/><i>25</i></span>}<br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_datastores_nfs").style.display =
"none";
    document.getElementById("more_datastores_nfs_button").style.display =
"none";
}

</script>

```

Edit the vars.yml file

The `vars.yml` file consolidates all environment-specific variables (ONTAP, Linux, or Oracle) for Oracle deployment.

- Edit and copy the variables from the VARS section and paste these variables into your `vars.yml` file.

VARS

```

<style>
div {
position: relative;
}
div button {
position: absolute;
top: 0;
right: 0;
}
button {
transition-duration: 0.4s;
background-color: white;
color: #1563a3;
border: 2px solid #1563a3;
}
button:hover {
background-color: #1563a3;
color: white;
}
#more_storage_vlans {
display: block;
}
#more_storage_vlans_button {
display: none;
}

```

```

#more_nfs_volumes {
    display: block;
}
#more_nfs_volumes_button {
    display: none;
}
</style>
<div class="listingblock"><div class="content"><div><button id="copy-
button" onclick="CopyClassText()">Copy</button></div><pre><code><div
class="CopyMeClass" id="CopyMeID">
#####
##### 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: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>ontap</i></span>

#CA_signed_certificates (ONLY CHANGE to 'true' IF YOU ARE USING CA SIGNED
CERTIFICATES)
ca_signed_certs: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"><i>false</i></span>

#Names of the Nodes in the ONTAP Cluster
nodes:
    - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"><i>AFF-
01</i></span>
    - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"><i>AFF-
02</i></span>

#Storage VLANs
#Add additional rows for vlans as necessary
storage_vlans:

```

```

- {vlan_id: ""<span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>203</i></span>", name: ""<span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>infra_NFS</i></span>",
protocol: ""<span <div
contenteditable="true""/><i>NFS</i></span>" }

<a id="more_storage_vlans" href="javascript:storagevlandropdown();">More
Storage VLANs</a><div id="select_more_storage_vlans"></div><a
id="more_storage_vlans_button" href="javascript:addstoragevlans();">Enter
Storage VLANs details</a><div id="extra_storage_vlans"></div>

#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: "<span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node01</i></span>"}
- {aggr_name: "<span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>aggr01_node02</i></span>"}

#SVM name
svm_name: "<span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>ora_svm</i></span>"

# SVM Management LIF Details
svm_mgmt_details:
- {address: "<span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>172.21.91.100</i></span>, netmask: "<span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"/><i>255.255.255.0</i></span>,
home_port: "<span <div contenteditable="true""/><i>e0M</i></span>"}

# 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 dedicated 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: ""<span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>{{groups.oracle[0]}}_u01</i></span>", aggr_name: <span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>aggr01_node01</i></span>, lif: <span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>172.21.94.200</i></span>, size: <span <div contenteditable='true'><i>25</i></span>}"
    - {vol_name: ""<span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>{{groups.oracle[0]}}_u02</i></span>", aggr_name: <span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>aggr01_node01</i></span>, lif: <span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>172.21.94.200</i></span>, size: <span <div contenteditable='true'><i>25</i></span>}"
      - {vol_name: ""<span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>{{groups.oracle[0]}}_u03</i></span>", aggr_name: <span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>aggr01_node01</i></span>, lif: <span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>172.21.94.200</i></span>, size: <span <div contenteditable='true'><i>25</i></span>}"
        <a id="more_nfs_volumes" href="javascript:nfsvolumesdropdown();">More NFS volumes</a><div id="select_more_nfs_volumes"></div><a id="more_nfs_volumes_button" href="javascript:addnfsvolumes();">Enter NFS volumes' details</a><div id="extra_nfs_volumes"></div>

#NFS LIFs IP address and netmask
nfs_lifs_details:
  - address: <span <div contenteditable='true' style='color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;'><i>172.21.94.200</i></span> #for node-1
    netmask: <span <div contenteditable='true' style='color:#004EFF; font-
```

```

weight:bold; font-style:italic; text-
decoration:underline;"/><i>255.255.255.0</i></span>
- address: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>172.21.94.201</i></span> #for node-2
    netmask: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>255.255.255.0</i></span>

#NFS client match
client_match: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>172.21.94.0/24</i></span>

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

#NFS Mount points for Oracle DB volumes
mount_points:
- <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u01</i></span>
- <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u02</i></span>
- <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"/><i>/u03</i></span>

# 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: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>1234</i></span>

# RedHat subscription username and password
redhat_sub_username: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>xxx</i></span>
redhat_sub_password: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"/><i>xxx</i></span>

```

```

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

db_domain: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>your.domain.com</i></span>

# Set initial password for all required Oracle passwords. Change them after installation.
initial_pwd_all: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>netapp123</i></span>

</div></code></pre></div></div>
<script>
function CopyClassText() {
    var textToCopy = document.getElementById("CopyMeID");
    var currentRange;
    if(document.getSelection().rangeCount > 0)
    {
        currentRange = document.getSelection().getRangeAt(0);
        window.getSelection().removeRange(currentRange);
    }
    else
    {
        currentRange = false;
    }
    var CopyRange = document.createRange();
    CopyRange.selectNode(textToCopy);
    window.getSelection().addRange(CopyRange);
    document.getElementById("more_storage_vlans").style.display = "none";
    document.getElementById("more_nfs_volumes").style.display = "none";
    var command = document.execCommand("copy");
    if (command)
    {
        document.getElementById("copy-button").innerHTML = "Copied!";
        setTimeout(revert_copy, 3000);
    }
    window.getSelection().removeRange(CopyRange);
    if(currentRange)
    {
        window.getSelection().addRange(currentRange);
    }
}
function revert_copy() {

```

```

        document.getElementById("copy-button").innerHTML = "Copy";
        document.getElementById("more_storage_vlans").style.display =
"block";
        document.getElementById("more_nfs_volumes").style.display = "block";
    }
    function storagevlandropdown() {
        document.getElementById("more_storage_vlans").style.display = "none";
        document.getElementById("more_storage_vlans_button").style.display =
"block";
        var x=1;
        var myHTML = '';
        var buildup = '';
        var wrapper = document.getElementById("select_more_storage_vlans");
        while (x < 10) {
            buildup += '<option value="' + x + '">' + x + '</option>';
            x++;
        }
        myHTML += '<a id="more_storage_vlans_info">How many extra VLANs do you
wish to add?</a><select name="number_of_extra_storage_vlans"
id="number_of_extra_storage_vlans">' + buildup + '</select>';
        wrapper.innerHTML = myHTML;
    }
    function addstoragevlans() {
        var y =
document.getElementById("number_of_extra_storage_vlans").value;
        var j=0;
        var myHTML = '';
        var wrapper = document.getElementById("extra_storage_vlans");
        while (j < y) {
            j++;
            myHTML += ' - {vlan_id: ""<span contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"><i>203</i></span>", name: ""<span contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"><i>infra_NFS</i></span>", protocol:
"&quot;<span contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"><i>NFS</i></span>&quot;}<br>';
        }
        wrapper.innerHTML = myHTML;
        document.getElementById("select_more_storage_vlans").style.display =
"none";
        document.getElementById("more_storage_vlans_button").style.display =
"none";
    }
    function nfsvolumesdropdown() {

```

```

document.getElementById("more_nfs_volumes").style.display = "none";
document.getElementById("more_nfs_volumes_button").style.display =
"block";
var x=1;
var myHTML = '';
var buildup = '';
var wrapper = document.getElementById("select_more_nfs_volumes");
while (x < 100) {
    buildup += '<option value="' + x + '">' + x + '</option>';
    x++;
}
myHTML += '<a id="more_nfs_volumes_info">How many extra NFS volumes do
you wish to add?</a><select name="number_of_extra_nfs_volumes"
id="number_of_extra_nfs_volumes">' + buildup + '</select>';
wrapper.innerHTML = myHTML;
}

function addnfsvolumes() {
var y = document.getElementById("number_of_extra_nfs_volumes").value;
var j=0;
var myHTML = '';
var wrapper = document.getElementById("extra_nfs_volumes");
while (j < y) {
    j++;
    myHTML += ' - {vol_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"><i>rtpora04_u01</i></span>, aggr_name: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"><i>aggr01_node02</i></span>,
lif: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-
decoration:underline;"><i>172.21.94.201</i></span>, size: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline;"><i>25</i></span>}<br>';
}
wrapper.innerHTML = myHTML;
document.getElementById("select_more_nfs_volumes").style.display =
"none";
document.getElementById("more_nfs_volumes_button").style.display =
"none";
}

</script>

```

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.

```
sqlplus / as sysdba
```

```
[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>
```

```
select name, log_mode from v$database;
```

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

```
NAME LOG_MODE
```

```
-----
```

```
CDB2 ARCHIVELOG
```

```
show pdbs;
```

```
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 the Oracle listener configuration with the following command. Change to the appropriate listener port and database service name.

```
sqlplus system@//localhost:1523/cdb2_pdb1.cie.netapp.com
```

[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 - netapp-lib - xmltodict - 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 - netapp-lib - xmltodict - 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 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

- Navigate to Resources → Credentials, and click Add.
- Enter the name and organization details for the ONTAP Credentials
- Select the credential type that was created in the previous step.
- 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.
- Click Save

4. Create Credential for Oracle (Source)

- Navigate to Resources → Credentials, and click Add.
- Enter the name and organization details for Oracle host
- Select the Machine credential type.
- Under Type Details, enter the Username and Password for the Oracle hosts.
- Select the correct Privilege Escalation Method, and enter the username and password.
- Click Save

5. Create Credential for Oracle Destination

- Navigate to Resources → Credentials, and click Add.
- Enter the name and organization details for the DR Oracle host
- Select the Machine credential type.
- Under Type Details, enter the Username (ec2-user or if you have changed it from default enter that), and the SSH Private Key
- Select the correct Privilege Escalation Method (sudo), and enter the username and password if needed.
- Click Save

Create a project

- 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

```
<style>
div {
position: relative;
}
div button {
position: absolute;
top: 0;
right: 0;
}
button {
transition-duration: 0.4s;
background-color: white;
color: #1563a3;
border: 2px solid #1563a3;
}
button:hover {
background-color: #1563a3;
color: white;
}
#more_binary_vols {
display: block;
}
#more_binary_vols_button {
display: none;
}
#more_database_vols {
display: block;
}
#more_database_vols_button {
display: none;
}
#more_log_vols {
display: block;
}
#more_log_vols_button {
display: none;
}
</style>
<div class="listingblock"><div class="content"><div><button
id="copy-button-onprem"
onclick="CopyClassText()">Copy</button></div><pre><code><div
class="CopyMeClass" id="CopyOnPrem">
#####
# #
#####
```

```

##### Oracle Data Protection global user configuration variables
#####
##### Consolidate all variables from ontap, aws, 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: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"><i>ontap</i></span>

#CA_signed_certificates (ONLY CHANGE to 'true' IF YOU ARE USING CA
SIGNED CERTIFICATES)
ca_signed_certs: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"><i>false</i></span>

#####
# Inter-cluster LIF details
#####
#Names of the Nodes in the Source ONTAP Cluster
src_nodes:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"><i>AFF-
01</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"><i>AFF-
02</i></span>

#Names of the Nodes in the Destination ONTAP Cluster
dst_nodes:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"><i>DR-
AFF-01</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"><i>DR-
AFF-02</i></span>

```

```

#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: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;" /><i>yes</i></span>

source_intercluster_network_port_details:
    using_dedicated_ports: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;" /><i>yes</i></span>
    using_ifgrp: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;" /><i>yes</i></span>
    using_vlans: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;" /><i>yes</i></span>
    failover_for_shared_individual_ports: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline; text-
decoration:underline;" /><i>yes</i></span>
    ifgrp_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;" /><i>a0a</i></span>
    vlan_id: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;" /><i>10</i></span>
    ports:
        - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;" /><i>e0b</i></span>
        - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;" /><i>e0g</i></span>
    broadcast_domain: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;" /><i>NFS</i></span>
    ipspace: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;" /><i>Default</i></span>
    failover_group_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;" /><i>iclifs</i></span>

source_intercluster_lif_details:

```

```

    - name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;" /><i>ic1_1</i></span>
        address: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;" /><i>10.0.0.1</i></span>
        netmask: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;" /><i>255.255.255.0</i></span>
        home_port: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;" /><i>a0a-
10</i></span>
        node: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;" /><i>AFF-01</i></span>
    - name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;" /><i>ic1_2</i></span>
        address: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;" /><i>10.0.0.2</i></span>
        netmask: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;" /><i>255.255.255.0</i></span>
        home_port: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;" /><i>a0a-
10</i></span>
        node: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;" /><i>AFF-02</i></span>

#Define whether or not to create intercluster lifs on destination
cluster (ONLY CHANGE to 'No' IF YOU HAVE ALREADY CREATED THE
INTERCLUSTER LIFS)
create_destination_intercluster_lifs: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline; text-
decoration:underline;" /><i>yes</i></span>

destination_intercluster_network_port_details:
    using_dedicated_ports: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;" /><i>yes</i></span>
    using_ifgrp: <span <div contenteditable="true"

```

```

        style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;"/><i>yes</i></span>
    using_vlans: <span <div contenteditable="true"
        style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;"/><i>yes</i></span>
    failover_for_shared_individual_ports: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;"/><i>yes</i></span>
    ifgrp_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline; text-decoration:underline;"/><i>a0a</i></span>
    vlan_id: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>10</i></span>
    ports:
        - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>e0b</i></span>
        - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>e0g</i></span>
    broadcast_domain: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>NFS</i></span>
    ipspace: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>Default</i></span>
    failover_group_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>iclifs</i></span>

destination_intercluster_lif_details:
    - name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>icl_1</i></span>
        address: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>10.0.0.3</i></span>
        netmask: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>255.255.255.0</i></span>
        home_port: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>a0a-

```

```

10</i></span>
    node: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"><i>DR-AFF-01</i></span>
      - name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"><i>icl_2</i></span>
        address: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"><i>10.0.0.4</i></span>
        netmask: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"><i>255.255.255.0</i></span>
        home_port: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"><i>a0a-
10</i></span>
        node: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"><i>DR-AFF-02</i></span>

#####
#####
# Variables for SnapMirror Peering
#####
#####
######
#src_lif: #Will be retrieve through Ansible Task
#dst_lif: #Will be retrieve through Ansible Task
passphrase: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"><i>your-passphrase</i></span>

#####
#####
#####
# Source & Destination List
#####
#####
#####
#Please Enter Destination Cluster Name
dst_cluster_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"><i>dst-cluster-
name</i></span>

#Please Enter Destination Cluster
dst_cluster_ip: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-

```

```

decoration:underline; text-decoration:underline;"/><i>dst-cluster-
ip</i></span>

#Please Enter Destination SVM to create mirror relationship
dst_vserver: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"><i>dst-vserver</i></span>

#Please Enter NFS Lif for dst vserver
dst_nfs_lif: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"><i>dst-nfs-lif</i></span>

#Please Enter Source Cluster Name
src_cluster_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"><i>src-cluster-
name</i></span>

#Please Enter Source Cluster
src_cluster_ip: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"><i>src-cluster-
ip</i></span>

#Please Enter Source SVM
src_vserver: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"><i>src-vserver</i></span>

#####
#####
#
# Variable for Oracle Volumes and SnapMirror Details
#####
#####
#
#Please Enter Source Snapshot Prefix Name
cg_snapshot_name_prefix: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"><i>oracle</i></span>

#Please Enter Source Oracle Binary Volume(s)
src_orabinary_vols:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"><i>binary_vol</i></span>

```

```

<a id="more_binary_vols"
    href="javascript:binaryvolsdropdown();">More Binary Vols</a><div
    id="select_more_binary_vols"></div><a id="more_binary_vols_button"
    href="javascript:addbinaryvols();">Enter Volume details</a><div
    id="extra_binary_vols"></div>
#Please Enter Source Database Volume(s)
src_db_vols:
    - <span <div contenteditable="true" style="color:#004EFF; font-
        weight:bold; font-style:italic; text-decoration:underline; text-
        decoration:underline;"><i>db_vol</i></span>
<a id="more_database_vols"
    href="javascript:databasevolsdropdown();">More Database Vols</a><div
    id="select_more_database_vols"></div><a
    id="more_database_vols_button"
    href="javascript:adddatabasevols();">Enter Volume details</a><div
    id="extra_database_vols"></div>
#Please Enter Source Archive Volume(s)
src_archivelog_vols:
    - <span <div contenteditable="true" style="color:#004EFF; font-
        weight:bold; font-style:italic; text-decoration:underline; text-
        decoration:underline;"><i>log_vol</i></span>
<a id="more_log_vols" href="javascript:logvolsdropdown();">More Log
Vols</a><div id="select_more_log_vols"></div><a
    id="more_log_vols_button" href="javascript:addlogvols();">Enter
    Volume details</a><div id="extra_log_vols"></div>
#Please Enter Destination Snapmirror Policy
snapmirror_policy: <span <div contenteditable="true"
    style="color:#004EFF; font-weight:bold; font-style:italic; text-
    decoration:underline; text-
    decoration:underline;"><i>async_policy_oracle</i></span>

#####
#####
# Export Policy Details
#####
#####
#Enter the destination export policy details
export_policy_details:
    name: <span <div contenteditable="true" style="color:#004EFF;
        font-weight:bold; font-style:italic; text-decoration:underline;
        text-decoration:underline;"><i>nfs_export_policy</i></span>
        client_match: <span <div contenteditable="true"
            style="color:#004EFF; font-weight:bold; font-style:italic; text-
            decoration:underline; text-
            decoration:underline;"><i>0.0.0.0/0</i></span>
        ro_rule: sys

```

```

rw_rule: sys

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

#NFS Mount points for Oracle DB volumes
mount_points:
  - <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>/u01</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>/u02</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>/u03</i></span>

# 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: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>1234</i></span>

# RedHat subscription username and password
redhat_sub_username: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>xxx</i></span>
redhat_sub_password: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>xxx</i></span>

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

#Recovery Type (leave as scn)
recovery_type: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>scn</i></span>

#Oracle Control Files
control_files:
  - <span <div contenteditable="true" style="color:#004EFF; font-

```

```

    weight:bold; font-style:italic; text-decoration:underline;"/><i>/u02/oradata/CDB2/control01.ctl</i></span>
    - <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"/><i>/u03/orareco/CDB2/control02.ctl</i></span>
    >

</div></code></pre></div></div>
<script>
function CopyClassText() {
    var textToCopy = document.getElementById("CopyOnPrem");
    var currentRange;
    if(document.getSelection().rangeCount > 0)
    {
        currentRange = document.getSelection().getRangeAt(0);
        window.getSelection().removeRange(currentRange);
    }
    else
    {
        currentRange = false;
    }
    var CopyRange = document.createRange();
    CopyRange.selectNode(textToCopy);
    window.getSelection().addRange(CopyRange);
    document.getElementById("more_binary_vols").style.display =
    "none";
    document.getElementById("more_database_vols").style.display =
    "none";
    document.getElementById("more_log_vols").style.display = "none";
    var command = document.execCommand("copy");
    if (command)
    {
        document.getElementById("copy-button-onprem").innerHTML =
        "Copied!";
        setTimeout(revert_copy, 3000);
    }
    window.getSelection().removeRange(CopyRange);
    if(currentRange)
    {
        window.getSelection().addRange(currentRange);
    }
}
function revert_copy() {
    document.getElementById("copy-button-onprem").innerHTML =
    "Copy";
}

```

```

        document.getElementById("more_binary_vols").style.display =
"block";
        document.getElementById("more_database_vols").style.display =
"block";
        document.getElementById("more_log_vols").style.display =
"block";
    }
    function binaryvolsdropdown() {
        document.getElementById("more_binary_vols").style.display =
"none";
        document.getElementById("more_binary_vols_button").style.display =
"block";
        var x=1;
        var myHTML = '';
        var buildup = '';
        var wrapper =
document.getElementById("select_more_binary_vols");
        while (x < 10) {
            buildup += '<option value="' + x + '">' + x + '</option>';
            x++;
        }
        myHTML += '<a id="more_binary_vols_info">How many extra volumes
do you wish to add?</a><select name="number_of_extra_binary_vols"
id="number_of_extra_binary_vols">' + buildup + '</select>';
        wrapper.innerHTML = myHTML;
    }
    function addbinaryvols() {
        var y =
document.getElementById("number_of_extra_binary_vols").value;
        var j=0;
        var myHTML = '';
        var wrapper = document.getElementById("extra_binary_vols");
        while (j < y) {
            j++;
            myHTML += ' - <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"><i>binary_vols</i></span><br>';
        }
        wrapper.innerHTML = myHTML;
        document.getElementById("select_more_binary_vols").style.display =
"none";
        document.getElementById("more_binary_vols_button").style.display =
"none";
    }
    function databasevolsdropdown() {

```

```

        document.getElementById("more_database_vols").style.display =
    "none";

document.getElementById("more_database_vols_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper =
document.getElementById("select_more_database_vols");
    while (x < 10) {
        buildup += '<option value="' + x + '">' + x + '</option>';
        x++;
    }
    myHTML += '<a id="more_database_vols_info">How many extra
volumes do you wish to add?</a><select
name="number_of_extra_database_vols"
id="number_of_extra_database_vols">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function adddatabasevols() {
    var y =
document.getElementById("number_of_extra_database_vols").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_database_vols");
    while (j < y) {
        j++;
        myHTML += ' - <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"><i>db_vol</i></span><br>';
    }
    wrapper.innerHTML = myHTML;

document.getElementById("select_more_database_vols").style.display =
"none";

document.getElementById("more_database_vols_button").style.display =
"none";
}

function logvolsdropdown() {
    document.getElementById("more_log_vols").style.display = "none";
    document.getElementById("more_log_vols_button").style.display =
"block";
    var x=1;

```

```

var myHTML = '';
var buildup = '';
var wrapper = document.getElementById("select_more_log_vols");
while (x < 10) {
    buildup += '<option value="' + x + '">' + x + '</option>';
    x++;
}
myHTML += '<a id="more_database_vols_info">How many extra
volumes do you wish to add?</a><select
name="number_of_extra_log_vols" id="number_of_extra_log_vols">' +
buildup + '</select>';
wrapper.innerHTML = myHTML;
}

function addlogvols() {
    var y =
document.getElementById("number_of_extra_log_vols").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_log_vols");
    while (j < y) {
        j++;
        myHTML += ' - <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"><i>log_vol</i></span><br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_log_vols").style.display =
"none";
    document.getElementById("more_log_vols_button").style.display =
"none";
}

</script>

```

CVO

```

<style>
div {
position: relative;
}
div button {
position: absolute;
top: 0;
right: 0;
}

```

```

button {
    transition-duration: 0.4s;
    background-color: white;
    color: #1563a3;
    border: 2px solid #1563a3;
}
button:hover {
    background-color: #1563a3;
    color: white;
}
#more_binary_vols1 {
    display: block;
}
#more_binary_vols1_button {
    display: none;
}
#more_database_vols1 {
    display: block;
}
#more_database_vols1_button {
    display: none;
}
#more_log_vols1 {
    display: block;
}
#more_log_vols1_button {
    display: none;
}

```

</style>

```

<div class="listingblock"><div class="content"><div><button
id="copy-button-cvo"
onclick="CopyClassTextCVO () ">Copy</button></div><pre><code><div
class="CopyMeClassCVO" id="CopyCVO">
#####
## Oracle Data Protection global user configuration variables
#####
##### Consolidate all variables from ontap, aws, CVO 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: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;" /><i>ontap</i></span>

#CA_signed_certificates (ONLY CHANGE to 'true' IF YOU ARE USING CA
SIGNED CERTIFICATES)
ca_signed_certs: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;" /><i>false</i></span>

#Names of the Nodes in the Source ONTAP Cluster
src_nodes:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;" /><i>AFF-
01</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;" /><i>AFF-
02</i></span>

#Names of the Nodes in the Destination CVO Cluster
dst_nodes:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;" /><i>DR-
AFF-01</i></span>
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;" /><i>DR-
AFF-02</i></span>

#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: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;" /><i>yes</i></span>

source_intercluster_network_port_details:
  using_dedicated_ports: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;" /><i>yes</i></span>
  using_ifgrp: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-

```

```

decoration:underline; text-decoration:underline;"/><i>yes</i></span>
using_vlans: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>yes</i></span>
failover_for_shared_individual_ports: <span <div
contenteditable="true" style="color:#004EFF; font-weight:bold; font-
style:italic; text-decoration:underline; text-
decoration:underline;"/><i>yes</i></span>
ifgrp_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>a0a</i></span>
vlan_id: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>10</i></span>
ports:
- <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>e0b</i></span>
- <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>e0g</i></span>
broadcast_domain: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>NFS</i></span>
ipspace: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>Default</i></span>
failover_group_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>iclifs</i></span>

source_intercluster_lif_details:
- name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>icl_1</i></span>
address: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>10.0.0.1</i></span>
netmask: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>255.255.255.0</i></span>
home_port: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>a0a-
10</i></span>

```

```

        node: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;" /><i>AFF-01</i></span>
      - name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;" /><i>icl_2</i></span>
        address: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;" /><i>10.0.0.2</i></span>
        netmask: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;" /><i>255.255.255.0</i></span>
        home_port: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;" /><i>a0a-
10</i></span>
        node: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;" /><i>AFF-02</i></span>

#####
### CVO Deployment Variables ###
#####

##### Access Keys Variables #####
# Region where your CVO will be deployed.
region_deploy: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;" /><i>us-east-1</i></span>

##### CVO and Connector Vars #####
# AWS Managed Policy required to give permission for IAM role
creation.
aws_policy: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;" /><i>arn:aws:iam::1234567:policy/OCCM</i></spa
n>

# Specify your aws role name, a new role is created if one already
does not exist.
aws_role_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;" /><i>arn:aws:iam::1234567:policy/OCCM</i></spa
n>
```

```

# Name your connector.
connector_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"><i>awx_connector</i></span>

# Name of the key pair generated in AWS.
key_pair: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"><i>key_pair</i></span>

# Name of the Subnet that has the range of IP addresses in your VPC.
subnet: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"><i>subnet-12345</i></span>

# ID of your AWS security group that allows access to on-prem
resources.
security_group: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline;"><i>sg-123123123</i></span>

# Your Cloud Manager Account ID.
account: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"><i>account-A23123A</i></span>

# Name of your CVO instance
cvo_name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-
decoration:underline;"><i>test_cvo</i></span>

# ID of the VPC in AWS.
vpc: <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline;"><i>vpc-
123123123</i></span>

#####
#####
# 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: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-

```

```

decoration:underline;"/><i>123123123123123123123</i></span>

# 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: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"><i>pass</i></span>

#####
#
# Variables for SnapMirror Peering
#####
#
passphrase: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"><i>your-passphrase</i></span>

#####
#
# Source & Destination List
#####
#
#Please Enter Destination Cluster Name
dst_cluster_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"><i>dst-cluster-
name</i></span>

#Please Enter Destination Cluster (Once CVO is Created Add this
Variable to all templates)
dst_cluster_ip: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"><i>dst-cluster-
ip</i></span>

#Please Enter Destination SVM to create mirror relationship
dst_vserver: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"><i>dst-vserver</i></span>

#Please Enter NFS Lif for dst vserver (Once CVO is Created Add this
Variable to all templates)
dst_nfs_lif: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;

```

```

text-decoration:underline;"/><i>dst-nfs-lif</i></span>

#Please Enter Source Cluster Name
src_cluster_name: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>src-cluster-
name</i></span>

#Please Enter Source Cluster
src_cluster_ip: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-decoration:underline;"/><i>src-cluster-
ip</i></span>

#Please Enter Source SVM
src_vserver: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"/><i>src-vserver</i></span>

#####
#####
#
# Variable for Oracle Volumes and SnapMirror Details
#####
#####
#####
#
# Please Enter Source Snapshot Prefix Name
cg_snapshot_name_prefix: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"/><i>oracle</i></span>

#Please Enter Source Oracle Binary Volume(s)
src_orabinary_vols:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>binary_vol</i></span>
<a id="more_binary_vols1"
href="javascript:binaryvols1dropdown();">More Binary Vols</a><div
id="select_more_binary_vols1"></div><a id="more_binary_vols1_button"
href="javascript:addbinaryvols1();">Enter Volume details</a><div
id="extra_binary_vols1"></div>
#Please Enter Source Database Volume(s)
src_db_vols:
  - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"/><i>db_vol</i></span>
<a id="more_database_vols1"

```

```

    href="javascript:databasevols1dropdown();">More Database
Vols</a><div id="select_more_database_vols1"></div><a
id="more_database_vols1_button"
href="javascript:adddatabasevols1();">Enter Volume details</a><div
id="extra_database_vols1"></div>
#Please Enter Source Archive Volume(s)
src_archivelog_vols:
    - <span <div contenteditable="true" style="color:#004EFF; font-
weight:bold; font-style:italic; text-decoration:underline; text-
decoration:underline;"><i>log_vol</i></span>
<a id="more_log_vols1" href="javascript:logvols1dropdown();">More
Log Vols</a><div id="select_more_log_vols1"></div><a
id="more_log_vols1_button" href="javascript:addlogvols1();">Enter
Volume details</a><div id="extra_log_vols1"></div>
#Please Enter Destination Snapmirror Policy
snapmirror_policy: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"><i>async_policy_oracle</i></span>

#####
#####
#
# Export Policy Details
#####
#####
#
#Enter the destination export policy details (Once CVO is Created
Add this Variable to all templates)
export_policy_details:
    name: <span <div contenteditable="true" style="color:#004EFF;
font-weight:bold; font-style:italic; text-decoration:underline;
text-decoration:underline;"><i>nfs_export_policy</i></span>
    client_match: <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"><i>0.0.0.0/0</i></span>
    ro_rule: sys
    rw_rule: sys

#####
#####
#
### Linux env specific config variables ###
#####
#####
#
#NFS Mount points for Oracle DB volumes
mount_points:

```

```

    - <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>/u01</i></span>
    - <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>/u02</i></span>
    - <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>/u03</i></span>

# 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: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>1234</i></span>

# RedHat subscription username and password
redhat_sub_username: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>xxx</i></span>
redhat_sub_password: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>xxx</i></span>

#####
### DB env specific install and config variables ##
#####
#Recovery Type (leave as scn)
recovery_type: <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>scn</i></span>

#Oracle Control Files
control_files:
    - <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>/u02/oradata/CDB2/control01.ctl</i></span>
    - <span <div contenteditable="true" style="color:#004EFF; font-weight:bold; font-style:italic; text-decoration:underline;"><i>/u03/orareco/CDB2/control02.ctl</i></span>

```

```

</div></code></pre></div></div>
<script>
function CopyClassTextCVO() {
    var textToCopy = document.getElementById("CopyCVO");
    var currentRange;
    if(document.getSelection().rangeCount > 0)
    {
        currentRange = document.getSelection().getRangeAt(0);
        window.getSelection().removeRange(currentRange);
    }
    else
    {
        currentRange = false;
    }
    var CopyRange = document.createRange();
    CopyRange.selectNode(textToCopy);
    window.getSelection().addRange(CopyRange);
    document.getElementById("more_binary_vols1").style.display =
"none";
    document.getElementById("more_database_vols1").style.display =
"none";
    document.getElementById("more_log_vols1").style.display =
"none";
    var command = document.execCommand("copy");
    if (command)
    {
        document.getElementById("copy-button-cvo").innerHTML =
"Copied!";
        setTimeout(revert_copy, 3000);
    }
    window.getSelection().removeRange(CopyRange);
    if(currentRange)
    {
        window.getSelection().addRange(currentRange);
    }
}
function revert_copy() {
    document.getElementById("copy-button-cvo").innerHTML = "Copy";
    document.getElementById("more_binary_vols1").style.display =
"block";
    document.getElementById("more_database_vols1").style.display =
"block";
    document.getElementById("more_log_vols1").style.display =
"block";
}
function binaryvols1dropdown() {

```

```

        document.getElementById("more_binary_vols1").style.display =
"none";

document.getElementById("more_binary_vols1_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper =
document.getElementById("select_more_binary_vols1");
    while (x < 10) {
        buildup += '<option value="' + x + '">' + x + '</option>';
        x++;
    }
    myHTML += '<a id="more_binary_vols1_info">How many extra volumes
do you wish to add?</a><select name="number_of_extra_binary_vols1"
id="number_of_extra_binary_vols1">' + buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function addbinaryvols1() {
    var y =
document.getElementById("number_of_extra_binary_vols1").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_binary_vols1");
    while (j < y) {
        j++;
        myHTML += ' - <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"><i>binary_vol</i></span><br>';
    }
    wrapper.innerHTML = myHTML;

document.getElementById("select_more_binary_vols1").style.display =
"none";
}

document.getElementById("more_binary_vols1_button").style.display =
"none";
}

function databasevols1dropdown() {
    document.getElementById("more_database_vols1").style.display =
"none";

document.getElementById("more_database_vols1_button").style.display =
"block";

```

```

var x=1;
var myHTML = '';
var buildup = '';
var wrapper =
document.getElementById("select_more_database_vols1");
while (x < 10) {
    buildup += '<option value="' + x + '">' + x + '</option>';
    x++;
}
myHTML += '<a id="more_database_vols1_info">How many extra
volumes do you wish to add?</a><select
name="number_of_extra_database_vols1"
id="number_of_extra_database_vols1">' + buildup + '</select>';
wrapper.innerHTML = myHTML;
}

function adddatabasevols1() {
    var y =
document.getElementById("number_of_extra_database_vols1").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_database_vols1");
    while (j < y) {
        j++;
        myHTML += ' - <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"><i>db_vol</i></span><br>';
    }
    wrapper.innerHTML = myHTML;

document.getElementById("select_more_database_vols1").style.display
= "none";

document.getElementById("more_database_vols1_button").style.display
= "none";
}
function logvols1dropdown() {
    document.getElementById("more_log_vols1").style.display =
"none";
    document.getElementById("more_log_vols1_button").style.display =
"block";
    var x=1;
    var myHTML = '';
    var buildup = '';
    var wrapper = document.getElementById("select_more_log_vols1");
    while (x < 10) {

```

```

        buildup += '<option value="' + x + '">' + x + '</option>';
        x++;
    }
    myHTML += '<a id="more_database_vols_info">How many extra
volumes do you wish to add?</a><select
name="number_of_extra_log_vols1" id="number_of_extra_log_vols1">' +
buildup + '</select>';
    wrapper.innerHTML = myHTML;
}
function addlogvols1() {
    var y =
document.getElementById("number_of_extra_log_vols1").value;
    var j=0;
    var myHTML = '';
    var wrapper = document.getElementById("extra_log_vols1");
    while (j < y) {
        j++;
        myHTML += ' - <span <div contenteditable="true"
style="color:#004EFF; font-weight:bold; font-style:italic; text-
decoration:underline; text-
decoration:underline;"><i>log_vol</i></span><br>';
    }
    wrapper.innerHTML = myHTML;
    document.getElementById("select_more_log_vols1").style.display =
"none";
    document.getElementById("more_log_vols1_button").style.display =
"none";
}

```

</script>

Automation Playbooks

There are four separate playbooks that need to be ran.

1. Playbook for Setting up your environment, On-Prem or CVO.
2. Playbook for replicating Oracle Binaries and Databases on a schedule
3. Playbook for replicating Oracle Logs on a schedule
4. Playbook for Recovering your database on a destination host

ONTAP/CVO Setup

ONTAP and CVO Setup

Configure and launch the job template.

1. Create the job template.
 - a. Navigate to Resources → Templates → Add and click Add Job Template.
 - b. Enter the name ONTAP/CVO Setup
 - c. Select the Job type; Run configures the system based on a playbook.
 - d. Select the corresponding inventory, project, playbook, and credentials for the playbook.
 - e. Select the ontap_setup.yml playbook for an On-Prem environment or select the cvo_setup.yml for replicating to a CVO instance.
 - f. Paste global variables copied from step 4 into the Template Variables field under the YAML tab.
 - g. Click Save.
2. Launch the job template.
 - a. Navigate to Resources → Templates.
 - b. Click the desired template and then click Launch.



We will use this template and copy it out for the other playbooks.

Replication For Binary and Database Volumes

Scheduling the Binary and Database Replication Playbook

Configure and launch the job template.

1. Copy the previously created job template.
 - a. Navigate to Resources → Templates.
 - b. Find the ONTAP/CVO Setup Template, and on the far right click on Copy Template
 - c. Click Edit Template on the copied template, and change the name to Binary and Database Replication Playbook.
 - d. Keep the same inventory, project, credentials for the template.
 - e. Select the ora_replication_cg.yml as the playbook to be executed.
 - f. The variables will remain the same, but the CVO cluster IP will need to be set in the variable dst_cluster_ip.
 - g. Click Save.
2. Schedule the job template.
 - a. Navigate to Resources → Templates.
 - b. Click the Binary and Database Replication Playbook template and then click Schedules at the top set of options.
 - c. Click Add, add Name Schedule for Binary and Database Replication, choose the Start date/time at the beginning of the hour, choose your Local time zone, and Run frequency. Run frequency will be often the SnapMirror replication will be updated.



A separate schedule will be created for the Log volume replication, so that it can be replicated on a more frequent cadence.

Replication for Log Volumes

Scheduling the Log Replication Playbook

Configure and launch the job template.

1. Copy the previously created job template.
 - a. Navigate to Resources → Templates.
 - b. Find the ONTAP/CVO Setup Template, and on the far right click on Copy Template
 - c. Click Edit Template on the copied template, and change the name to Log Replication Playbook.
 - d. Keep the same inventory, project, credentials for the template.
 - e. Select the ora_replication_logs.yml as the playbook to be executed.
 - f. The variables will remain the same, but the CVO cluster IP will need to be set in the variable dst_cluster_ip.
 - g. Click Save.
2. Schedule the job template.
 - a. Navigate to Resources → Templates.
 - b. Click the Log Replication Playbook template and then click Schedules at the top set of options.
 - c. Click Add, add Name Schedule for Log Replication, choose the Start date/time at the beginning of the hour, choose your Local time zone, and Run frequency. Run frequency will be often the SnapMirror replication will be updated.



It is recommended to set the log schedule to update every hour to ensure the recovery to the last hourly update.

Restore and Recover Database

Scheduling the Log Replication Playbook

Configure and launch the job template.

1. Copy the previously created job template.
 - a. Navigate to Resources → Templates.
 - b. Find the ONTAP/CVO Setup Template, and on the far right click on Copy Template
 - c. Click Edit Template on the copied template, and change the name to Restore and Recovery Playbook.
 - d. Keep the same inventory, project, credentials for the template.
 - e. Select the ora_recovery.yml as the playbook to be executed.
 - f. The variables will remain the same, but the CVO cluster IP will need to be set in the variable dst_cluster_ip.
 - g. Click Save.



This playbook will not be ran until you are ready to restore your database at the remote site.

Recovering Oracle Database

1. On-premises production Oracle databases data volumes are protected via NetApp SnapMirror replication to either a redundant ONTAP cluster in secondary data center or Cloud Volume ONTAP in public cloud. In a fully configured disaster recovery environment, recovery compute instances in secondary data center or public cloud are standby and ready to recover the production database in the case of a disaster. The standby compute instances are kept in sync with on-prem instances by running parallel updates on OS kernel patch or upgrade in a lockstep.
2. In this solution demonstrated, Oracle binary volume is replicated to target and mounted at target instance to bring up Oracle software stack. This approach to recover Oracle has advantage over a fresh installation of Oracle at last minute when a disaster occurred. It guarantees Oracle installation is fully in sync with current on-prem production software installation and patch levels etc. However, this may or may not have additional software licensing implication for the replicated Oracle binary volume at recovery site depending on how the software licensing is structured with Oracle. User is recommended to check with its software licensing personnel to assess the potential Oracle licensing requirement before deciding to use the same approach.
3. The standby Oracle host at the destination is configured with the Oracle prerequisite configurations.
4. The SnapMirrors are broken and the volumes are made writable and mounted to the standby Oracle host.
5. The Oracle recovery module performs following tasks to recovery and startup Oracle at recovery site after all DB volumes are mounted at standby compute instance.
 - a. Sync the control file: We deployed duplicate Oracle control files on different database volume to protect critical database control file. One is on the data volume and another is on log volume. Since data and log volumes are replicated at different frequency, they will be out of sync at the time of recovery.
 - b. Relink Oracle binary: Since the Oracle binary is relocated to a new host, it needs a relink.
 - c. Recover Oracle database: The recovery mechanism retrieves last System Change Number in last available archived log in Oracle log volume from control file and recovers Oracle database to recoup all business transactions that was able to be replicated to DR site at the time of failure. The database is then started up in a new incarnation to carry on user connections and business transaction at recovery site.



Before running the Recovering playbook make sure you have the following:
Make sure it copy over the /etc/oratab and /etc/oralInst.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](#)

Copyright information

Copyright © 2023 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—with prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.