# TR-5003: High Throughput Oracle VLDB Implementation on ANF

## NetApp database solutions

NetApp
August 05, 2025

# Table of Contents


TR-5003: High Throughput Oracle VLDB Implementation on ANF . . . 1
- Purpose . . . 1
- Audience . . . 1
- Solution test and validation environment . . . 1
  - Architecture . . . 2
  - Hardware and software components . . . 2
  - Oracle VLDB Data Guard configuration with a simulated NY to LA DR setup . . . 3
  - Key factors for deployment consideration . . . 3
- Solution deployment . . . 3
  - Prerequisites for deployment . . . 4
  - Primary Oracle VLDB configuration for Data Guard . . . 7
  - Standby Oracle VLDB configuration for Data Guard . . . 16
  - Setup Data Guard Broker . . . 25
  - Clone standby database for other use cases via automation . . . 28
- Where to find additional information . . . 29

# TR-5003: High Throughput Oracle VLDB Implementation on ANF

Allen Cao, Niyaz Mohamed, NetApp

The solution provides an overview and details for configuring a high-throughput Oracle Very Large Database (VLDB) on Microsoft Azure NetApp Files (ANF) with Oracle Data Guard in the Azure cloud.

## Purpose

High throughput and mission-critical Oracle VLDB put a heavy demand on backend database storage. To meet service level agreement (SLA), the database storage must deliver the required capacity and high input/output operations per second (IOPS) while maintaining sub milliseconds latency performance. This is particularly challenging when deploying such a database workload in the public cloud with a shared storage resources environment. Not all storage platforms are created equal. Premium Azure NetApp Files storage in combination with Azure infrastructure can meet the needs of such a highly demanding Oracle workload. In a validated performance benchmark (Oracle database performance on Azure NetApp Files multiple volumes), ANF delivered 2.5 million read IOPS with 700 microseconds latency in a synthetic 100% random select workload via the SLOB tool. With a standard 8k block size, this translates to about 20 GiB/s throughput.

In this documentation, we demonstrate how to set up an Oracle VLDB with Data Guard configuration on ANF storage with multiple NFS volumes and Oracle ASM for storage load balancing. The standby database can be quickly (mins) backed up via snapshot and cloned for read/write access for use cases as desired. NetApp Solutions Engineering team provides an automation toolkit to create and refresh clones with ease at an user defined schedule.

This solution addresses the following use cases:

- Implementation of Oracle VLDB in a Data Guard setting on Microsoft Azure NetApp Files storage across Azure regions.
- Snapshot backup and clone the physical standby database to serve use cases such as reporting, dev, test, etc. via automation.

## Audience

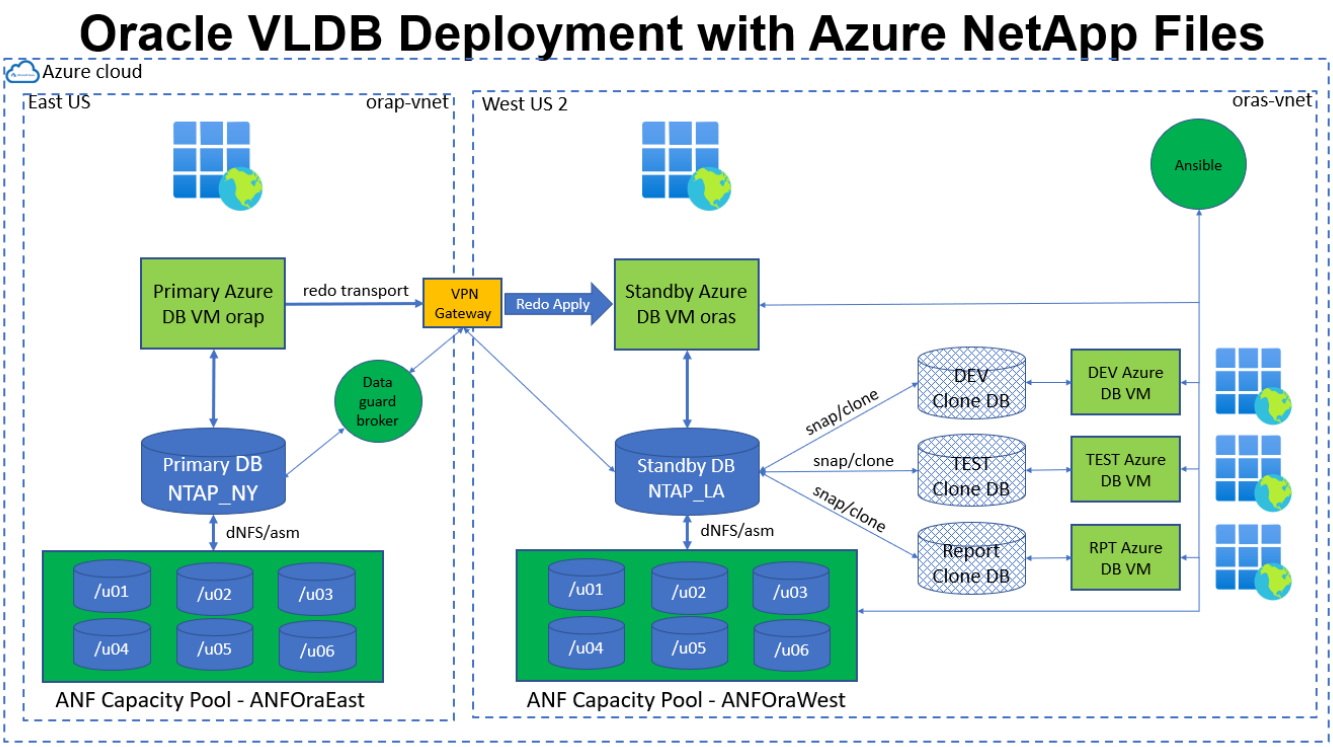
This solution is intended for the following people:

- A DBA who sets up Oracle VLDB with Data Guard in Azure cloud for high availability, data protection, and disaster recovery.
- A database solution architect interested in Oracle VLDB with Data Guard configuration in the Azure cloud.
- A storage administrator who manages Azure NetApp Files storage that supports Oracle database.
- An application owner who likes to stand up Oracle VLDB with Data Guard in an Azure cloud environment.

## Solution test and validation environment

The testing and validation of this solution was performed in an Azure cloud lab setting that might not match the actual user deployment environment. For more information, see the section Key factors for deployment consideration.

## Architecture




## Hardware and software components

| | | |
|---|---|---|
| **Hardware** | | |
| Azure NetApp Files | Current version offered by Microsoft | Two 4 TiB Capacity Pools, Premium Service Level, Auto QoS |
| Azure VMs for DB Servers | Standard B4ms (4 vcpus, 16 GiB memory) | Three DB VMs, one as the primary DB server, one as the standby DB server, and the third as a clone DB server |
| **Software** | | |
| RedHat Linux | Red Hat Enterprise Linux 8.6 (LVM) - x64 Gen2 | 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 |
| dNFS OneOff Patch | p32931941_190000_Linux-x86-64.zip | Applied to both grid and database |
| Oracle OPatch | Version 12.2.0.1.36 | Latest patch p6880880_190000_Linux-x86-64.zip |

| | | |
|---|---|---|
| Ansible | Version core 2.16.2 | python version - 3.10.13 |
| NFS | Version 3.0 | dNFS enabled for Oracle |

### Oracle VLDB Data Guard configuration with a simulated NY to LA DR setup

| Database | DB_UNIQUE_NAME | Oracle Net Service Name |
|---|---|---|
| Primary | NTAP_NY | NTAP_NY.internal.cloudapp.net |
| Standby | NTAP_LA | NTAP_LA.internal.cloudapp.net |

### Key factors for deployment consideration

- **Azure NetApp Files Configuration.** Azure NetApp Files are allocated in the Azure NetApp storage account as `Capacity Pools`. In these tests and validations, we deployed a 2 TiB capacity pool to host Oracle primary at the East region and a 4 TiB capacity pool to host standby database and DB clone at the West 2 region. ANF capacity pool has three service levels: Standard, Premium, and Ultra. The IO capacity of ANF capacity pool is based on the size of the capacity pool and its service level. At a capacity pool creation, you can set QoS to Auto or Manual and data encryption at rest Single or Double.
- **Sizing the Database Volumes.** For production deployment, NetApp recommends taking a full assessment of your Oracle database throughput requirement from Oracle AWR report. Take into consideration both the database size as well as the throughput requirements when sizing ANF volumes for database. With auto QoS configuration for ANF, the bandwidth is guaranteed at 128 MiB/s per TiB volume capacity allocated with Ultra Service Level. Higher throughput may need larger volume sizing to meet the requirement.
- **Single Volume or Multiple Volumes.** A single large volume can provide similar performance level as multiple volumes with same aggregate size as the QoS is strictly enforced based on the volume sizing and capacity pool service level. It is recommended to implement multiple volumes (multiple NFS mount points) for Oracle VLDB to better utilize shared backend ANF storage resource pool. Implement Oracle ASM for IO load balancing on multiple NFS volumes.
- **Application Volume Group.** Deploy Application Volume Group (AVG) for Oracle for performance optimization. Volumes deployed by application volume group are placed in the regional or zonal infrastructure to achieve optimized latency and throughput for the application VMs.
- **Azure VM Consideration.** In these tests and validations, we used an Azure VM - Standard_B4ms with 4 vCPUs and16 GiB memory. You need to choose the Azure DB VM appropriately for Oracle VLDB with high throughput requirement. Besides the number of vCPUs and the amount of RAM, the VM network bandwidth (ingress and egress or NIC throughput limit) can become a bottleneck before database storage capacity is reached.
- **dNFS Configuration.** By using dNFS, an Oracle database running on an Azure Virtual Machine with ANF storage can drive significantly more I/O than the native NFS client. Ensure that Oracle dNFS patch p32931941 is applied to address potential bugs.

# Solution deployment

It is assumed that you already have your primary Oracle database deployed in an Azure cloud environment within a VNet as the starting point for setting up the Oracle Data Guard. Ideally, the primary database is deployed on ANF storage with NFS mount. Your primary Oracle database can also be running on a NetApp ONTAP storage or any other storage of choices either within the Azure ecosystem or a private data center. The following section demonstrates the configuration for Oracle VLDB on ANF in an Oracle Data Guard setting between a primary Oracle DB in Azure with ANF storage to a physical standby Oracle DB in Azure with ANF

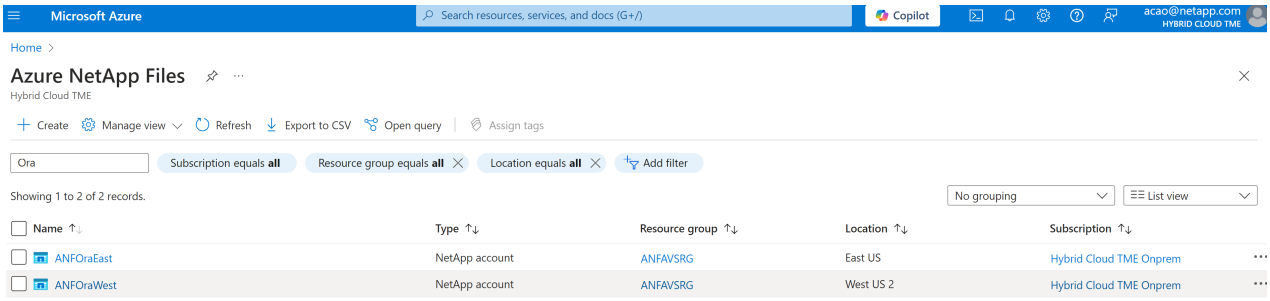
storage.

## Prerequisites for deployment

Deployment requires the following prerequisites.

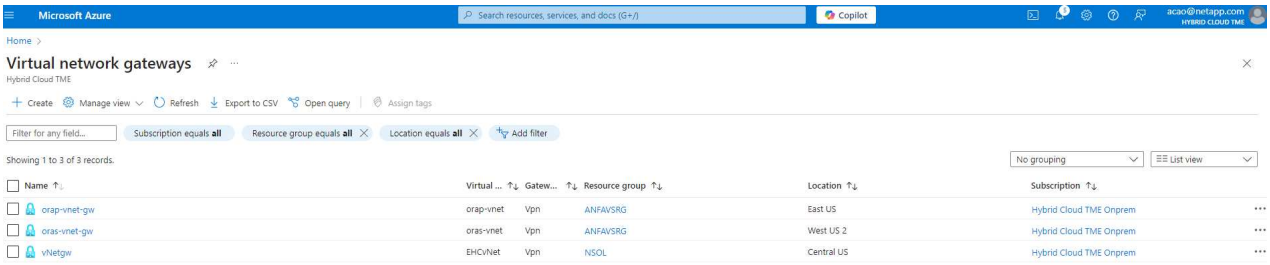
1. An Azure cloud account has been set up, and the necessary VNet and network subnets have been created within your Azure account.
2. From the Azure cloud portal console, you need to deploy minimum three Azure Linux VMs, one as the primary Oracle DB server, one as the standby Oracle DB server, and a clone target DB server for reporting, dev, and test etc. See the architecture diagram in the previous section for more details about the environment setup. Also review the Microsoft Azure Virtual Machines for more information.
3. The primary Oracle database should have been installed and configured in the primary Oracle DB server. On the other hand, in the standby Oracle DB server or the clone Oracle DB server, only Oracle software is installed and no Oracle databases are created. Ideally, the Oracle files directories layout should be exactly matching on all Oracle DB servers. For details on NetApp recommendation for automated Oracle deployment in the Azure cloud and ANF, please refer to the following technical reports for help.
   - TR-4987: Simplified, Automated Oracle Deployment on Azure NetApp Files with NFS

   Ensure that you have allocated at least 128G in the Azure VMs root volume in order to have sufficient space to stage Oracle installation files.
4. From the Azure cloud portal console, deploy two ANF storage capacity pools to host Oracle database volumes. The ANF storage capacity pools should be situated in different regions to mimic a true DataGuard configuration. If you are not familiar with the deployment of ANF storage, see the documentation Quickstart: Set up Azure NetApp Files and create an NFS volume for step-by-step instructions.

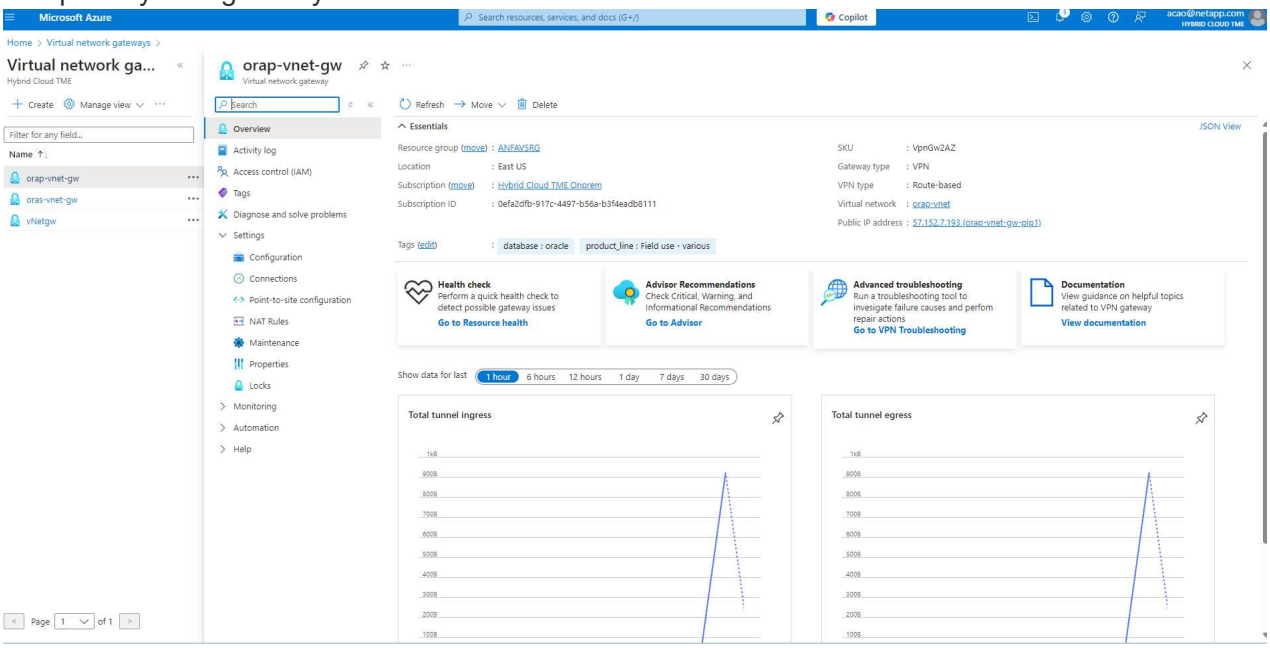



5. When the primary Oracle database and the standby Oracle database are situated in two different regions, a VPN gateway should be configured to allow data traffic flow between two separate VNets. Detailed networking configuration in Azure is beyond the scope of this document. Following screen shots provides some reference on how the VPN gateways are configured, connected, and the data traffics flow are confirmed in the lab.

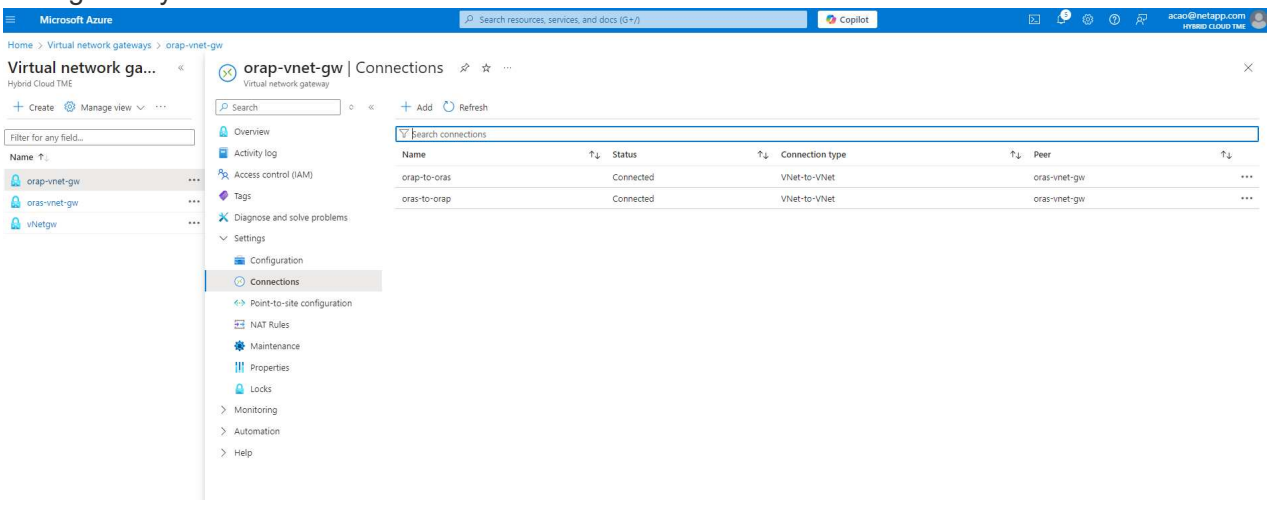
   Lab VPN gateways:



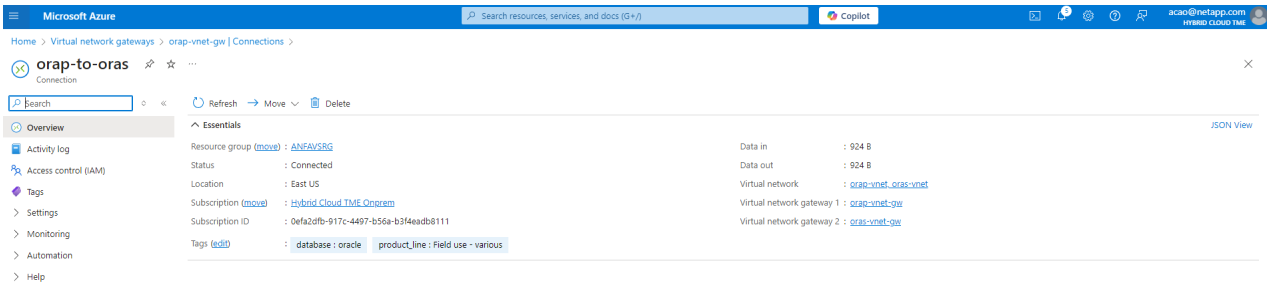

The primary vnet gateway:




Vnet gateway connection status:




Validate that the traffic flows are established (click on three dots to open the page):




6. Refer to this documentation Deploy application volume group for Oracle to deploy Application Volume Group for Oracle.

## Primary Oracle VLDB configuration for Data Guard

In this demonstration, we have setup a primary Oracle database called NTAP on the primary Azure DB server with six NFS mount points: /u01 for the Oracle binary, /u02, /u04, /u05, /u06 for the Oracle data files, and an Oracle control file, /u03 for the Oracle active logs, archived log files, and a redundant Oracle control file. This setup serves as a reference configuration. Your actual deployment should take into consideration of your specific needs and requirements in terms of the capacity pool sizing, the service level, the number of database volumes and the sizing of each volume.

For detailed step by step procedures for setting up Oracle Data Guard on NFS with ASM, please refer to TR-5002- Oracle Active Data Guard Cost Reduction with Azure NetApp Files and TR-4974- Oracle 19c in Standalone Restart on AWS FSx/EC2 with NFS/ASM relevant sections. Although the procedures in TR-4974 were validated on Amazon FSx ONTAP, they are equally applicable to ANF. Following illustrates the details of a primary Oracle VLDB in a Data Guard configuration.

1. The primary database NTAP on the primary Azure DB server orap.internal.cloudapp.net is initially deployed as a standalone database with the ANF on NFS and ASM as database storage.

```
orap.internal.cloudapp.net:
resource group: ANFAVSRG
Location: East US
size: Standard B4ms (4 vcpus, 16 GiB memory)
OS: Linux (redhat 8.6)
pub_ip: 172.190.207.231
pri_ip: 10.0.0.4

[oracle@orap ~]$ df -h
Filesystem                 Size  Used Avail Use% Mounted on
devtmpfs                   7.7G     0  7.7G   0% /dev
tmpfs                      7.8G  1.1G  6.7G  15% /dev/shm
tmpfs                      7.8G   17M  7.7G   1% /run
tmpfs                      7.8G     0  7.8G   0% /sys/fs/cgroup
/dev/mapper/rootvg-rootlv   22G   20G  2.1G  91% /
/dev/mapper/rootvg-usrlv    10G  2.3G  7.8G  23% /usr
/dev/sda1                  496M  181M  315M  37% /boot
/dev/mapper/rootvg-varlv   8.0G  1.1G  7.0G  13% /var
/dev/sda15                 495M  5.8M  489M   2% /boot/efi
/dev/mapper/rootvg-homelv  2.0G   47M  2.0G   3% /home
/dev/mapper/rootvg-tmplv    12G   11G  1.9G  85% /tmp
/dev/sdb1                   32G   49M   30G   1% /mnt
10.0.2.38:/orap-u06        300G  282G   19G  94% /u06
10.0.2.38:/orap-u04        300G  282G   19G  94% /u04
10.0.2.36:/orap-u01        400G   21G  380G   6% /u01
10.0.2.37:/orap-u02        300G  282G   19G  94% /u02
10.0.2.36:/orap-u03        400G  282G  119G  71% /u03
10.0.2.39:/orap-u05        300G  282G   19G  94% /u05


[oracle@orap ~]$ cat /etc/oratab
```

```
#



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

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

2. Login to primary DB server as the oracle user. Validate grid configuration.

```
$GRID_HOME/bin/crsctl stat res -t
```

```
[oracle@orap ~]$ $GRID_HOME/bin/crsctl stat res -t
--------------------------------------------------------------------------------
------------
Name           Target  State        Server                   State
details
--------------------------------------------------------------------------------
------------
Local Resources
--------------------------------------------------------------------------------
------------
ora.DATA.dg
               ONLINE  ONLINE       orap                     STABLE
ora.LISTENER.lsnr
               ONLINE  ONLINE       orap                     STABLE
ora.LOGS.dg
               ONLINE  ONLINE       orap                     STABLE
ora.asm
               ONLINE  ONLINE       orap
Started,STABLE
ora.ons
               OFFLINE OFFLINE      orap                     STABLE
--------------------------------------------------------------------------------
------------
Cluster Resources
--------------------------------------------------------------------------------
------------
ora.cssd
      1        ONLINE  ONLINE       orap                     STABLE
ora.diskmon
      1        OFFLINE OFFLINE                               STABLE
ora.evmd
      1        ONLINE  ONLINE       orap                     STABLE
ora.ntap.db
      1        OFFLINE OFFLINE
Instance Shutdown,ST
                                                             ABLE
--------------------------------------------------------------------------------
------------
[oracle@orap ~]$
```

3. ASM disk group configuration.

```
asmcmd
```

```
[oracle@orap ~]$ asmcmd
ASMCMD> lsdg
State    Type    Rebal  Sector  Logical_Sector  Block       AU
Total_MB  Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks
Voting_files  Name
MOUNTED  EXTERN  N         512             512   4096  4194304
1146880  1136944                0         1136944              0
N  DATA/
MOUNTED  EXTERN  N         512             512   4096  4194304
286720   283312                0          283312              0
N  LOGS/
ASMCMD> lsdsk
Path
/u02/oradata/asm/orap_data_disk_01
/u02/oradata/asm/orap_data_disk_02
/u02/oradata/asm/orap_data_disk_03
/u02/oradata/asm/orap_data_disk_04
/u03/oralogs/asm/orap_logs_disk_01
/u03/oralogs/asm/orap_logs_disk_02
/u03/oralogs/asm/orap_logs_disk_03
/u03/oralogs/asm/orap_logs_disk_04
/u04/oradata/asm/orap_data_disk_05
/u04/oradata/asm/orap_data_disk_06
/u04/oradata/asm/orap_data_disk_07
/u04/oradata/asm/orap_data_disk_08
/u05/oradata/asm/orap_data_disk_09
/u05/oradata/asm/orap_data_disk_10
/u05/oradata/asm/orap_data_disk_11
/u05/oradata/asm/orap_data_disk_12
/u06/oradata/asm/orap_data_disk_13
/u06/oradata/asm/orap_data_disk_14
/u06/oradata/asm/orap_data_disk_15
/u06/oradata/asm/orap_data_disk_16
ASMCMD>
```

4. Parameters setting for Data Guard on primary DB.

```
SQL> show parameter name

NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
cdb_cluster_name                     string
cell_offloadgroup_name               string
db_file_name_convert                 string
```

```
db_name                              string      NTAP
db_unique_name                       string      NTAP_NY
global_names                         boolean     FALSE
instance_name                        string      NTAP
lock_name_space                      string
log_file_name_convert                string
pdb_file_name_convert                string
processor_group_name                 string

NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
service_names                        string
NTAP_NY.internal.cloudapp.net

SQL> sho parameter log_archive_dest

NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
log_archive_dest                     string
log_archive_dest_1                   string
LOCATION=USE_DB_RECOVERY_FILE_
                                                 DEST
VALID_FOR=(ALL_LOGFILES,A
                                                 LL_ROLES)
DB_UNIQUE_NAME=NTAP_
                                                 NY
log_archive_dest_10                  string
log_archive_dest_11                  string
log_archive_dest_12                  string
log_archive_dest_13                  string
log_archive_dest_14                  string
log_archive_dest_15                  string

NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
log_archive_dest_16                  string
log_archive_dest_17                  string
log_archive_dest_18                  string
log_archive_dest_19                  string
log_archive_dest_2                   string      SERVICE=NTAP_LA
ASYNC VALID_FO

R=(ONLINE_LOGFILES,PRIMARY_ROL
```

```
                                                    E)
DB_UNIQUE_NAME=NTAP_LA
log_archive_dest_20                  string
log_archive_dest_21                  string
log_archive_dest_22                  string
```

5. Primary DB configuration.

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

NAME      OPEN_MODE            LOG_MODE
--------- -------------------- ------------
NTAP      READ WRITE           ARCHIVELOG


SQL> show pdbs

    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 NTAP_PDB1                      READ WRITE NO
         4 NTAP_PDB2                      READ WRITE NO
         5 NTAP_PDB3                      READ WRITE NO


SQL> select name from v$datafile;

NAME
------------------------------------------------------------------------
------------
+DATA/NTAP/DATAFILE/system.257.1189724205
+DATA/NTAP/DATAFILE/sysaux.258.1189724249
+DATA/NTAP/DATAFILE/undotbs1.259.1189724275
+DATA/NTAP/86B637B62FE07A65E053F706E80A27CA/DATAFILE/system.266.1189
725235
+DATA/NTAP/86B637B62FE07A65E053F706E80A27CA/DATAFILE/sysaux.267.1189
725235
+DATA/NTAP/DATAFILE/users.260.1189724275
+DATA/NTAP/86B637B62FE07A65E053F706E80A27CA/DATAFILE/undotbs1.268.11
89725235
+DATA/NTAP/2B1302C26E089A59E0630400000A4D5C/DATAFILE/system.272.1189
726217
+DATA/NTAP/2B1302C26E089A59E0630400000A4D5C/DATAFILE/sysaux.273.1189
726217
+DATA/NTAP/2B1302C26E089A59E0630400000A4D5C/DATAFILE/undotbs1.271.11
89726217
```

```
+DATA/NTAP/2B1302C26E089A59E0630400000A4D5C/DATAFILE/users.275.11897
26243

NAME
--------------------------------------------------------------------
------------
+DATA/NTAP/2B13047FB98B9AAFE0630400000AFA5F/DATAFILE/system.277.1189
726245
+DATA/NTAP/2B13047FB98B9AAFE0630400000AFA5F/DATAFILE/sysaux.278.1189
726245
+DATA/NTAP/2B13047FB98B9AAFE0630400000AFA5F/DATAFILE/undotbs1.276.11
89726245
+DATA/NTAP/2B13047FB98B9AAFE0630400000AFA5F/DATAFILE/users.280.11897
26269
+DATA/NTAP/2B13061057039B10E0630400000AA001/DATAFILE/system.282.1189
726271
+DATA/NTAP/2B13061057039B10E0630400000AA001/DATAFILE/sysaux.283.1189
726271
+DATA/NTAP/2B13061057039B10E0630400000AA001/DATAFILE/undotbs1.281.11
89726271
+DATA/NTAP/2B13061057039B10E0630400000AA001/DATAFILE/users.285.11897
26293

19 rows selected.

SQL> select member from v$logfile;

MEMBER
--------------------------------------------------------------------
------------
+DATA/NTAP/ONLINELOG/group_3.264.1189724351
+LOGS/NTAP/ONLINELOG/group_3.259.1189724361
+DATA/NTAP/ONLINELOG/group_2.263.1189724351
+LOGS/NTAP/ONLINELOG/group_2.257.1189724359
+DATA/NTAP/ONLINELOG/group_1.262.1189724351
+LOGS/NTAP/ONLINELOG/group_1.258.1189724359
+DATA/NTAP/ONLINELOG/group_4.286.1190297279
+LOGS/NTAP/ONLINELOG/group_4.262.1190297283
+DATA/NTAP/ONLINELOG/group_5.287.1190297293
+LOGS/NTAP/ONLINELOG/group_5.263.1190297295
+DATA/NTAP/ONLINELOG/group_6.288.1190297307

MEMBER
--------------------------------------------------------------------
------------
+LOGS/NTAP/ONLINELOG/group_6.264.1190297309
```

```
+DATA/NTAP/ONLINELOG/group_7.289.1190297325
+LOGS/NTAP/ONLINELOG/group_7.265.1190297327

14 rows selected.

SQL> select name from v$controlfile;

NAME
--------------------------------------------------------------------------------
------------
+DATA/NTAP/CONTROLFILE/current.261.1189724347
+LOGS/NTAP/CONTROLFILE/current.256.1189724347
```

6. dNFS configuration on primary DB.

```
SQL> select svrname, dirname from v$dnfs_servers;

SVRNAME
------------------------------------------------------------------
------------
DIRNAME
------------------------------------------------------------------
------------
10.0.2.39
/orap-u05

10.0.2.38
/orap-u04

10.0.2.38
/orap-u06


SVRNAME
------------------------------------------------------------------
------------
DIRNAME
------------------------------------------------------------------
------------
10.0.2.37
/orap-u02

10.0.2.36
/orap-u03

10.0.2.36
/orap-u01


6 rows selected.
```

This completes the demonstration of a Data Guard setup for VLDB NTAP at the primary site on ANF with NFS/ASM.

## Standby Oracle VLDB configuration for Data Guard

Oracle Data Guard requires OS kernel configuration and Oracle software stacks including patch sets on standby DB server to match with primary DB server. For easy management and simplicity, the database storage configuration of the standby DB server ideally should match with the primary DB server as well, such as the database directory layout and sizes of NFS mount points.

Again, for detailed step by step procedures for setting up Oracle Data Guard standby on NFS with ASM, please refer to TR-5002 - Oracle Active Data Guard Cost Reduction with Azure NetApp Files and TR-4974 - Oracle 19c in Standalone Restart on AWS FSx/EC2 with NFS/ASM relevant sections. Following illustrates the detail of standby Oracle VLDB configuration on standby DB server in a Data Guard setting.

1. The standby Oracle DB server configuration at standby site in the demo lab.

```
oras.internal.cloudapp.net:
resource group: ANFAVSRG
Location: West US 2
size: Standard B4ms (4 vcpus, 16 GiB memory)
OS: Linux (redhat 8.6)
pub_ip: 172.179.119.75
pri_ip: 10.0.1.4

[oracle@oras ~]$ df -h
Filesystem                 Size  Used Avail Use% Mounted on
devtmpfs                   7.7G     0  7.7G   0% /dev
tmpfs                      7.8G  1.1G  6.7G  15% /dev/shm
tmpfs                      7.8G   25M  7.7G   1% /run
tmpfs                      7.8G     0  7.8G   0% /sys/fs/cgroup
/dev/mapper/rootvg-rootlv   22G   17G  5.6G  75% /
/dev/mapper/rootvg-usrlv    10G  2.3G  7.8G  23% /usr
/dev/mapper/rootvg-varlv   8.0G  1.1G  7.0G  13% /var
/dev/mapper/rootvg-homelv  2.0G   52M  2.0G   3% /home
/dev/sda1                  496M  181M  315M  37% /boot
/dev/sda15                 495M  5.8M  489M   2% /boot/efi
/dev/mapper/rootvg-tmplv    12G   11G  1.8G  86% /tmp
/dev/sdb1                   32G   49M   30G   1% /mnt
10.0.3.36:/oras-u03        400G  282G  119G  71% /u03
10.0.3.36:/oras-u04        300G  282G   19G  94% /u04
10.0.3.36:/oras-u05        300G  282G   19G  94% /u05
10.0.3.36:/oras-u02        300G  282G   19G  94% /u02
10.0.3.36:/oras-u01        100G   21G   80G  21% /u01
10.0.3.36:/oras-u06        300G  282G   19G  94% /u06

[oracle@oras ~]$ cat /etc/oratab
#Backup file is
/u01/app/oracle/crsdata/oras/output/oratab.bak.oras.oracle line
added by Agent
#
```

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

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

2. Grid infrastructure configuration on standby DB server.

```
[oracle@oras ~]$ $GRID_HOME/bin/crsctl stat res -t
--------------------------------------------------------------------------------
------------
Name           Target  State        Server                   State
details
--------------------------------------------------------------------------------
------------
Local Resources
--------------------------------------------------------------------------------
------------
ora.DATA.dg
               ONLINE  ONLINE       oras                     STABLE
ora.LISTENER.lsnr
               ONLINE  ONLINE       oras                     STABLE
ora.LOGS.dg
               ONLINE  ONLINE       oras                     STABLE
ora.asm
               ONLINE  ONLINE       oras
Started,STABLE
ora.ons
               OFFLINE OFFLINE      oras                     STABLE
--------------------------------------------------------------------------------
------------
Cluster Resources
--------------------------------------------------------------------------------
------------
ora.cssd
      1        ONLINE  ONLINE       oras                     STABLE
ora.diskmon
      1        OFFLINE OFFLINE                               STABLE
ora.evmd
      1        ONLINE  ONLINE       oras                     STABLE
ora.ntap_la.db
      1        ONLINE  INTERMEDIATE oras
Dismounted,Mount Ini

tiated,HOME=/u01/app

/oracle/product/19.0

.0/NTAP,STABLE
--------------------------------------------------------------------------------
------------
```

3. ASM disk groups configuration on standby DB server.

```
[oracle@oras ~]$ asmcmd
ASMCMD> lsdg
State    Type    Rebal  Sector  Logical_Sector  Block       AU
Total_MB  Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks
Voting_files  Name
MOUNTED  EXTERN  N         512             512   4096  4194304
1146880  1136912                0         1136912              0
N  DATA/
MOUNTED  EXTERN  N         512             512   4096  4194304
286720   284228                0          284228              0
N  LOGS/
ASMCMD> lsdsk
Path
/u02/oradata/asm/oras_data_disk_01
/u02/oradata/asm/oras_data_disk_02
/u02/oradata/asm/oras_data_disk_03
/u02/oradata/asm/oras_data_disk_04
/u03/oralogs/asm/oras_logs_disk_01
/u03/oralogs/asm/oras_logs_disk_02
/u03/oralogs/asm/oras_logs_disk_03
/u03/oralogs/asm/oras_logs_disk_04
/u04/oradata/asm/oras_data_disk_05
/u04/oradata/asm/oras_data_disk_06
/u04/oradata/asm/oras_data_disk_07
/u04/oradata/asm/oras_data_disk_08
/u05/oradata/asm/oras_data_disk_09
/u05/oradata/asm/oras_data_disk_10
/u05/oradata/asm/oras_data_disk_11
/u05/oradata/asm/oras_data_disk_12
/u06/oradata/asm/oras_data_disk_13
/u06/oradata/asm/oras_data_disk_14
/u06/oradata/asm/oras_data_disk_15
/u06/oradata/asm/oras_data_disk_16
```

4. Parameters setting for Data Guard on standby DB.

```
SQL> show parameter name

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

NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
service_names                        string
NTAP_LA.internal.cloudapp.net
SQL> show parameter log_archive_config

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

NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
fal_server                           string      NTAP_NY
```

5. Standby DB configuration.

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

NAME      OPEN_MODE            LOG_MODE
--------- -------------------- ------------
NTAP      MOUNTED              ARCHIVELOG

SQL> show pdbs
```

```
    CON_ID CON_NAME                        OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       MOUNTED
         3 NTAP_PDB1                      MOUNTED
         4 NTAP_PDB2                      MOUNTED
         5 NTAP_PDB3                      MOUNTED

SQL> select name from v$datafile;

NAME
--------------------------------------------------------------------------------
------------
+DATA/NTAP_LA/DATAFILE/system.261.1190301867
+DATA/NTAP_LA/DATAFILE/sysaux.262.1190301923
+DATA/NTAP_LA/DATAFILE/undotbs1.263.1190301969
+DATA/NTAP_LA/2B12C97618069248E0630400000AC50B/DATAFILE/system.264.1
190301987
+DATA/NTAP_LA/2B12C97618069248E0630400000AC50B/DATAFILE/sysaux.265.1
190302013
+DATA/NTAP_LA/DATAFILE/users.266.1190302039
+DATA/NTAP_LA/2B12C97618069248E0630400000AC50B/DATAFILE/undotbs1.267
.1190302045
+DATA/NTAP_LA/2B1302C26E089A59E0630400000A4D5C/DATAFILE/system.268.1
190302071
+DATA/NTAP_LA/2B1302C26E089A59E0630400000A4D5C/DATAFILE/sysaux.269.1
190302099
+DATA/NTAP_LA/2B1302C26E089A59E0630400000A4D5C/DATAFILE/undotbs1.270
.1190302125
+DATA/NTAP_LA/2B1302C26E089A59E0630400000A4D5C/DATAFILE/users.271.11
90302133

NAME
--------------------------------------------------------------------------------
------------
+DATA/NTAP_LA/2B13047FB98B9AAFE0630400000AFA5F/DATAFILE/system.272.1
190302137
+DATA/NTAP_LA/2B13047FB98B9AAFE0630400000AFA5F/DATAFILE/sysaux.273.1
190302163
+DATA/NTAP_LA/2B13047FB98B9AAFE0630400000AFA5F/DATAFILE/undotbs1.274
.1190302189
+DATA/NTAP_LA/2B13047FB98B9AAFE0630400000AFA5F/DATAFILE/users.275.11
90302197
+DATA/NTAP_LA/2B13061057039B10E0630400000AA001/DATAFILE/system.276.1
190302201
+DATA/NTAP_LA/2B13061057039B10E0630400000AA001/DATAFILE/sysaux.277.1
```

```
190302229
+DATA/NTAP_LA/2B13061057039B10E0630400000AA001/DATAFILE/undotbs1.278
.1190302255
+DATA/NTAP_LA/2B13061057039B10E0630400000AA001/DATAFILE/users.279.11
90302263

19 rows selected.

SQL> select name from v$controlfile;

NAME
--------------------------------------------------------------------
------------
+DATA/NTAP_LA/CONTROLFILE/current.260.1190301831
+LOGS/NTAP_LA/CONTROLFILE/current.257.1190301833

SQL> select group#, type, member from v$logfile order by 2, 1;
    GROUP# TYPE    MEMBER
---------- -------
--------------------------------------------------------------------
------------
         1 ONLINE  +DATA/NTAP_LA/ONLINELOG/group_1.280.1190302305
         1 ONLINE  +LOGS/NTAP_LA/ONLINELOG/group_1.259.1190302309
         2 ONLINE  +DATA/NTAP_LA/ONLINELOG/group_2.281.1190302315
         2 ONLINE  +LOGS/NTAP_LA/ONLINELOG/group_2.258.1190302319
         3 ONLINE  +DATA/NTAP_LA/ONLINELOG/group_3.282.1190302325
         3 ONLINE  +LOGS/NTAP_LA/ONLINELOG/group_3.260.1190302329
         4 STANDBY +DATA/NTAP_LA/ONLINELOG/group_4.283.1190302337
         4 STANDBY +LOGS/NTAP_LA/ONLINELOG/group_4.261.1190302339
         5 STANDBY +DATA/NTAP_LA/ONLINELOG/group_5.284.1190302347
         5 STANDBY +LOGS/NTAP_LA/ONLINELOG/group_5.262.1190302349
         6 STANDBY +DATA/NTAP_LA/ONLINELOG/group_6.285.1190302357

    GROUP# TYPE    MEMBER
---------- -------
--------------------------------------------------------------------
------------
         6 STANDBY +LOGS/NTAP_LA/ONLINELOG/group_6.263.1190302359
         7 STANDBY +DATA/NTAP_LA/ONLINELOG/group_7.286.1190302367
         7 STANDBY +LOGS/NTAP_LA/ONLINELOG/group_7.264.1190302369

14 rows selected.
```

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

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

ROLE                        THREAD#  SEQUENCE# ACTION
--------------------------- ---------- ---------- ------------
recovery logmerger                   1         32 APPLYING_LOG
recovery apply slave                 0          0 IDLE
RFS async                            1         32 IDLE
recovery apply slave                 0          0 IDLE
recovery apply slave                 0          0 IDLE
RFS ping                             1         32 IDLE
archive redo                         0          0 IDLE
managed recovery                     0          0 IDLE
archive redo                         0          0 IDLE
archive redo                         0          0 IDLE
recovery apply slave                 0          0 IDLE

ROLE                        THREAD#  SEQUENCE# ACTION
--------------------------- ---------- ---------- ------------
redo transport monitor               0          0 IDLE
log writer                           0          0 IDLE
archive local                        0          0 IDLE
redo transport timer                 0          0 IDLE
gap manager                          0          0 IDLE
RFS archive                          0          0 IDLE

17 rows selected.
```

7. dNFS configuration on standby DB.

```
SQL> select svrname, dirname from v$dnfs_servers;

SVRNAME
--------------------------------------------------------------------------------
---------
DIRNAME
--------------------------------------------------------------------------------
---------
10.0.3.36
/oras-u05

10.0.3.36
/oras-u04

10.0.3.36
/oras-u02

10.0.3.36
/oras-u06

10.0.3.36
/oras-u03
```

This completes the demonstration of a Data Guard setup for VLDB NTAP with managed standby recovery enabled at standby site.

## Setup Data Guard Broker

Oracle Data Guard broker is a distributed management framework that automates and centralizes the creation, maintenance, and monitoring of Oracle Data Guard configurations. Following section demonstrate how to setup Data Guard Broker to manage Data Guard environment.

1. Start data guard broker on both the primary and the standby databases with following command via sqlplus.

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

2. From primary database, connect to Data Guard Borker as SYSDBA.

```
[oracle@orap ~]$ dgmgrl sys@NTAP_NY
DGMGRL for Linux: Release 19.0.0.0.0 - Production on Wed Dec 11
20:53:20 2024
Version 19.18.0.0.0

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

Welcome to DGMGRL, type "help" for information.
Password:
Connected to "NTAP_NY"
Connected as SYSDBA.
DGMGRL>
```

3. Create and enable Data Guard Broker configuration.

```
DGMGRL> create configuration dg_config as primary database is
NTAP_NY connect identifier is NTAP_NY;
Configuration "dg_config" created with primary database "ntap_ny"
DGMGRL> add database NTAP_LA as connect identifier is NTAP_LA;
Database "ntap_la" added
DGMGRL> enable configuration;
Enabled.
DGMGRL> show configuration;

Configuration - dg_config

  Protection Mode: MaxPerformance
  Members:
  ntap_ny - Primary database
    ntap_la - Physical standby database

Fast-Start Failover:  Disabled

Configuration Status:
SUCCESS   (status updated 3 seconds ago)
```

4. Validate the database status within the Data Guard Broker management framework.

```
DGMGRL> show database db1_ny;

Database - db1_ny

  Role:               PRIMARY
  Intended State:     TRANSPORT-ON
  Instance(s):
    db1

Database Status:
SUCCESS

DGMGRL> show database db1_la;

Database - db1_la

  Role:               PHYSICAL STANDBY
  Intended State:     APPLY-ON
  Transport Lag:      0 seconds (computed 1 second ago)
  Apply Lag:          0 seconds (computed 1 second ago)
  Average Apply Rate: 2.00 KByte/s
  Real Time Query:    OFF
  Instance(s):
    db1

Database Status:
SUCCESS

DGMGRL>
```

In the event of a failure, Data Guard Broker can be used to failover the primary database to the standby instantaniouly. If `Fast-Start Failover` is enabled, Data Guard Broker can failover the primary database to the standby when a failure is detected without an user intervention.

## Clone standby database for other use cases via automation

Following automation toolkit is specifically designed to create or refresh clones of an Oracle Data Guard standby DB deployed to ANF with NFS/ASM configuration for a complete clone lifecycle management.

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

The toolkit can only be accessed by NetApp internal user with bitbucket access at this moment. For interested external users, please request access from your account team or reach out to NetApp Solutions Engineering team.

# Where to find additional information

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

- TR-5002: Oracle Active Data Guard Cost Reduction with Azure NetApp Files

  TR-5002: Oracle Active Data Guard Cost Reduction with Azure NetApp Files

- TR-4974: Oracle 19c in Standalone Restart on AWS FSx/EC2 with NFS/ASM

  TR-4974: Oracle 19c in Standalone Restart on AWS FSx/EC2 with NFS/ASM

- Azure NetApp Files

  https://azure.microsoft.com/en-us/products/netapp

- Oracle Data Guard Concepts and Administration

  https://docs.oracle.com/en/database/oracle/oracle-database/19/sbydb/index.html#Oracle%C2%AE-Data-Guard

## Copyright information



## Trademark information