Deploy a Hybrid Data Guard on Oracle Database in OCI (ExaC@C/ExaDB-D/Base/DB on IaaS) using the Recover Standby Database from Service clause

Local/Cross Region Disaster Recovery

By:
Rob Watson rob.watson@oracle.com
Sneha Nitin Pednekar sneha.pednekar@oracle.com

DISCLAIMER

Get Started

Maintaining business continuity and ensuring IT resiliency is a top priority for IT leaders today. Enterprises from every sector and industry are increasingly implementing Disaster Recovery solutions to benefit from best-in-class services, competitive pricing, agility, flexibility, and higher availability while enhancing risk management.

In this document, we will be emulating a hybrid data guard for disaster recovery scenario where the primary is a Base database and the standby is a Base database in OCI.
This can be used for demos, POCs and workshops in conceptualizing the integrity and safety of the data in case of a disaster in OCI environment. This document does not in any way supersede any existing whitepapers referenced in the documentation, it only builds on it.

Considerations for deploying a Disaster Recovery topology using OCI for Oracle Database

Before configuring the DR topology, gather the information that you'll need to set up the standby database.

1. Determine the size and CIDR block of the virtual cloud network (VCN) that you want to create, and the DNS label of the VCN for Primary and Standby in OCI. See Allowed VCN Size and Address Ranges.
2. Verify that the service limits of your tenancy can accommodate all the resources that you want to create.
3. Determine the DB VM system display names of Primary and Standby.
4. Determine the DB Unique names of Primary and Standby.
5. The database names must be the same both dbs, unique names has to be different.
6. Obtain the DB admin password for your database.
7. Save the path to the public SSH key.
8. Save the path to the private SSH key.

Roles needed for each service.

Service Name: Role	Required to...
Oracle Database: root, opc, oracle	Configure the primary database and instantiate and configure the standby database.
Oracle Cloud Infrastructure: sysdba	Close, shutdown, and unmount the standby database in the cloud.
Oracle Data Guard: SYS, SYSDG or SYSDBA	Run the Oracle Data Guard command-line interface (DGMGRL)

Prerequisites

1. Create Oracle Database in OCI

Create the standby database using UI with the same database name and instance name. Choose the same database version and edition for both the databases.

2. ssh key

Ensure you have the ssh private key for the Oracle databases. Test the connectivity for the Primary and Standby Oracle databases.

3. Network Setup for DB Systems in OCI

Configure /etc/hosts on Primary and Standby environment.
Edit tnsnames.ora file for both Oracle databases to have a connection to the "other" database. Then confirm that the tnsping is working fine from both DB systems to the opposite database.

In case the standby is being created in a cross region, set the following prior to the editing tnsnames.ora file with Oracle database entries:
Set up a Remote Peering Connection between the two Regions in OCI. This is done through the respective DRGs. Establish a remote peering connection between the existing DRGs. Peer Status should be "Peered". Allow traffic between the VCNs in the 2 regions.
Navigate to the VCN details, and select the default security list. Then select "Add Ingress Rules" to allow traffic from the opposite VCN's CIDR range.

References

Ensure to understand the considerations/requirements/perquisites mentioned in the documents below:
Base Database - https://docs.oracle.com/en/cloud/paas/base-database/data-guard/index.html#articletitle
ExaDB-D – https://docs.oracle.com/en/engineered-systems/exadata-cloud-service/ecscm/using-data-guard-with-exacc.html#GUID-6EBC4D6A-C58B-4721-B756-F22FC6819A45
ExaC@C – https://docs.oracle.com/en/engineered-systems/exadata-cloud-at-customer/ecccm/ecc-using-data-guard.html#GUID-6EBC4D6A-C58B-4721-B756-F22FC6819A45

If the environment is On-premise, refer to the document below to setup the primary database.

https://docs.oracle.com/en/solutions/standby-database-in-cloud/encrypt-data-source-database1.html#GUID-5F306AB7-1489-4934-860E-C846206ABF8D

Environment

The current environment setup is as below:

PRIMARY

- Base Database VM - sndel
- DBNAME - sndel
- UNIQUE NAME - sndel_p

STANDBY

- Base Database VM - sndel
- DBNAME - sndel
- UNIQUE NAME - sndel_s

Prepare the primary database

To prepare the primary database, you'll need to configure static listeners, update tnsnames.ora, and configure some database settings and parameters, like:

- Check database flashback is enabled.
- Check force database logging is enabled.
- Check database is in archive log mode.
- Check database is in open mode.
- Check database is in Primary database role.

The output will look similar to the following:

```
SQL> SELECT LOG_MODE, FORCE_LOGGING, FLASHBACK_ON, OPEN_MODE, DATABASE_ROLE FROM V$DATABASE;

LOG_MODE        FORCE_LOGGING  FLASHBACK_ON   OPEN_MODE          DATABASE_ROLE
--------------- -------------- -------------- ------------------ ----------------
ARCHIVELOG      YES            YES            READ WRITE         PRIMARY
```

The output will look similar to the following:

```
SQL> show parameter standby_file_management

NAME                                 TYPE        VALUE
------------------------------------ ----------- ----------
standby_file_management              string      AUTO
```

- Broker configuration files should be set

The output will look similar to the following:

```
SQL> show parameter dg_broker_config_file1;
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
dg_broker_config_file1               string      /u01/app/oracle/product/19.0.0.0/dbhome_1/dbs/dr1dbuks_898_lhr.dat


SQL> show parameter dg_broker_config_file2;

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
dg_broker_config_file2               string      /u01/app/oracle/product/19.0.0.0/dbhome_1/dbs/dr2dbuks_898_lhr.da
```

- Enable Data Guard Broker

The output will look similar to the following:

```
SQL> show parameter dg_broker_start
NAME                                 TYPE        VALUE
------------------------------------ ----------- --------------
dg_broker_start                      boolean     FALSE

SQL> alter system set dg_broker_start=true scope=both;
SQL> show parameter dg_broker_start

NAME                                 TYPE        VALUE
------------------------------------ ----------- --------------
dg_broker_start                      boolean     TRUE
```

- Check listener status.

The output will look similar to the following:

```
[oracle@sndel tde]$ lsnrctl status

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 06-MAR-2024 13:40:48
Copyright (c) 1991, 2023, Oracle.  All rights reserved.
Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
--------------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 19.0.0.0.0 - Production
Start Date                06-MAR-2024 12:09:54
Uptime                    0 days 1 hr. 30 min. 54 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /u01/app/19.0.0.0/grid/network/admin/listener.ora
Listener Log File         /u01/app/grid/diag/tnslsnr/sndel/listener/alert/log.xml
Listening Endpoints Summary...
```

```
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=10.0.0.48)(PORT=1521)))
Services Summary...
Service "+APX" has 1 instance(s).
  Instance "+APX1", status READY, has 1 handler(s) for this service...
Service "+ASM" has 1 instance(s).
  Instance "+ASM1", status READY, has 1 handler(s) for this service...
Service "+ASM_DATA" has 1 instance(s).
  Instance "+ASM1", status READY, has 1 handler(s) for this service...
Service "+ASM_RECO" has 1 instance(s).
  Instance "+ASM1", status READY, has 1 handler(s) for this service...
Service "02a979ca24549a30e0630706f40ab807.sub06291309370.snvcn.oraclevcn.com" has 1 instance(s).
  Instance "sndel", status READY, has 2 handler(s) for this service...
Service "12fe1204497c090fe0633000000ab26f.sub06291309370.snvcn.oraclevcn.com" has 1 instance(s).
  Instance "sndel", status READY, has 2 handler(s) for this service...
Service "goa.sub06291309370.snvcn.oraclevcn.com" has 1 instance(s).
  Instance "sndel", status READY, has 2 handler(s) for this service...
Service "sndelXDB.sub06291309370.snvcn.oraclevcn.com" has 1 instance(s).
  Instance "sndel", status READY, has 1 handler(s) for this service...
Service "sndel_goa.paas.oracle.com" has 1 instance(s).
  Instance "sndel", status READY, has 2 handler(s) for this service...
Service "sndel_p.sub06291309370.snvcn.oraclevcn.com" has 1 instance(s).
  Instance "sndel", status READY, has 2 handler(s) for this service...
The command completed successfully
```

- Note the tns entries (Update Primary with Standby tns entry and Update Standby with Primary tns entry)

```
[oracle@sndel ]$  cat $ORACLE_HOME/network/admin/tnsnames.ora
# tnsnames.ora Network Configuration File:
/u01/app/oracle/product/19.0.0.0/dbhome_1/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.
SNDEL_P =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = sndel)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = sndel_p.sub06291309370.snvcn.oraclevcn.com)
    )
  )

SNDEL_S =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = sndeldr)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = sndel_s.sub06291309370.snvcn.oraclevcn.com)
    )
  )


LISTENER_SNDEL =
  (ADDRESS = (PROTOCOL = TCP)(HOST = sndel)(PORT = 1521))
```

- Note the sqlnet.ora output.

```
[oracle@sndel ]$  cat $ORACLE_HOME/network/admin/sqlnet.ora
```

- Note the database configuration details.

```
[oracle@sndel ]$  srvctl config database -d <db_unique_name>
```

The output will look similar to the following:

```
[oracle@sndel tde]$ srvctl config database -d sndel_p
Database unique name: sndel_p
Database name: sndel
Oracle home: /u01/app/oracle/product/19.0.0.0/dbhome_1
Oracle user: oracle
Spfile: +DATA/SNDEL_P/PARAMETERFILE/spfile.262.1162902271
Password file:
Domain: sub06291309370.snvcn.oraclevcn.com
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Server pools:
Disk Groups: RECO,DATA
Mount point paths: /opt/oracle/dcs/commonstore
Services: sndel_goa.paas.oracle.com
Type: SINGLE
OSDBA group: dba
OSOPER group: dbaoper
Database instance: sndel
Configured nodes: sndel
CSS critical: no
CPU count: 0
Memory target: 0
Maximum memory: 0
Default network number for database services:
Database is administrator managed
[oracle@sndel tde]$
```

- Copy the Password File to the standby DB system in the /tmp location

```
[oracle@~]$ cd $ORACLE_HOME/dbs
[oracle@~]$ cp $ORACLE_HOME/dbs/orapw<sid> /tmp/orapw<sid>
[oracle@~]$ chmod 777 /tmp/orapw<sid>
[opc@ ~]$ cd .ssh
[opc@ ~]$ scp -i <private_key> /tmp/orapw<sid> opc@<ip_standby_vm>:/tmp/orapw<sid>
```

The output will look similar to the following:

```
[oracle@sndel tde]$ cd $ORACLE_HOME/dbs
[oracle@sndel dbs]$ ls -ltra
-rw-r--r--  1 oracle oinstall 3079 May 14  2015 init.ora
-rw-rw----  1 oracle asmadmin 1544 Mar  6 12:28 hc_sndel.dat
-rw-r-----  1 oracle oinstall 2560 Mar  6 13:03 orapwsndel

[oracle@sndel dbs]$ cp $ORACLE_HOME/dbs/orapwsndel /tmp
chmod 777 /tmp/orapwsndel
scp -i <private_key> /tmp/orapwsndel opc@<ip_standby_vm>:/tmp/orapwsndel
```

- Locate and copy the wallet files to the standby DB system in the /tmp location

SQL> select CON_ID, WRL_PARAMETER, WRL_TYPE, STATUS, WALLET_TYPE from V$ENCRYPTION_WALLET;

[oracle@ ~]$ cd /opt/oracle/dcs/commonstore/wallets/<db_unique_name>/tde

[oracle@ tde]$ ls -ltra
-rw------- 1 oracle asmadmin 5467 Jun 19 18:59 ewallet.p12
-rw------- 1 oracle asmadmin 5512 Jun 19 18:59 cwallet.sso

[oracle@ ~]$ cp /opt/oracle/dcs/commonstore/wallets/<db_unique_name>/tde/ewallet.p12 /tmp/ewallet.p12
[oracle@ ~]$ cp /opt/oracle/dcs/commonstore/wallets/< db_unique_name>/tde/cwallet.sso /tmp/cwallet.sso

[oracle@ ~]$ chmod 777 /tmp/ewallet.p12
[oracle@ ~]$ chmod 777 /tmp/cwallet.sso

[opc@.ssh]$ scp -i <private_key> /tmp/ewallet.p12 opc@<ip_standby_vm>:/tmp/ewallet.p12
[opc@.ssh]$ scp -i <private_key> /tmp/cwallet.sso opc@<ip_standby_vm>:/tmp/cwallet.sso

The output will look similar to the following:

CON_ID	WRL_PARAMETER	STATUS	WALLET_TYPE
1	/opt/oracle/dcs/commonstore/wallets/sndel_p/tde/	OPEN	AUTOLOGIN
2		OPEN	AUTOLOGIN
3		OPEN	AUTOLOGIN

Wallet files to scp to dr site –

[oracle@ ~]$ cd /opt/oracle/dcs/commonstore/wallets/sndel_p/tde/

-rw------- 1 oracle asmadmin 5467 Mar 6 12:26 ewallet.p12
-rw------- 1 oracle asmadmin 5512 Mar 6 12:26 cwallet.sso

[oracle@ ~]$ cp /opt/oracle/dcs/commonstore/wallets/sndel_p /tde/ewallet.p12 /tmp/ewallet.p12
[oracle@ ~]$ cp /opt/oracle/dcs/commonstore/wallets/sndel_p/tde/cwallet.sso /tmp/cwallet.sso

[oracle@ ~]$ chmod 777 /tmp/ewallet.p12
[oracle@ ~]$ chmod 777 /tmp/cwallet.sso

[opc@.ssh]$ scp -i <private_key> /tmp/ewallet.p12 opc@<ip_standby_vm>:/tmp/ewallet.p12
[opc@.ssh]$ scp -i <private_key> /tmp/cwallet.sso opc@<ip_standby_vm>:/tmp/cwallet.sso

Prepare the standby database

- Manually delete the database created by OCI UI

Before deleting the Standby Database, save the current db_unique_name and note the wallet location as that will be used later. Below query provides details for the wallet location. *(For DB on IaaS the standby database will not be created initially by OCI UI and the instance can be initiated with a init.ora file similar to on-premise DG creation).*

SQL> select CON_ID, WRL_PARAMETER, WRL_TYPE, STATUS, WALLET_TYPE from V$ENCRYPTION_WALLET;

The entry will look similar to the following:

SQL> select CON_ID, WRL_PARAMETER, WRL_TYPE, STATUS, WALLET_TYPE from V$ENCRYPTION_WALLET;

CON_ID	WRL_PARAMETER	STATUS	WALLET_TYPE
1	/opt/oracle/dcs/commonstore/wallets/sndel_s/tde/	OPEN	AUTOLOGIN
2		OPEN	AUTOLOGIN
3		OPEN	AUTOLOGIN

- Note the database configuration details

```
srvctl config database -d <db_unique_name>
```

The entry will look similar to the following:

[oracle@sndeldr ~]$ srvctl config database -d sndel_s
Database unique name: sndel_s
Database name: sndel
Oracle home: /u01/app/oracle/product/19.0.0.0/dbhome_1
Oracle user: oracle
Spfile: +DATA/SNDEL_S/PARAMETERFILE/spfile.262.1162906943
Password file:
Domain: sub06291309370.snvcn.oraclevcn.com
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Server pools:
Disk Groups: RECO,DATA
Mount point paths: /opt/oracle/dcs/commonstore
Services: sndel_sndel_pdb1.paas.oracle.com
Type: SINGLE
OSDBA group: dba
OSOPER group: dbaoper
Database instance: sndel
Configured nodes: sndeldr
CSS critical: no
CPU count: 0
Memory target: 0
Maximum memory: 0
Default network number for database services:
Database is administrator managed

- Remove the database files and shutdown the database which is being built as a Standby

```
vi rm_dbfiles.sql
set heading off linesize 999 pagesize 0 feedback off trimspool on
spool /home/oracle/demo/files.lst
select 'asmcmd rm '||name from v$datafile union all select 'asmcmd rm '||name from
v$tempfile union all select 'asmcmd rm '||member from v$logfile;
spool off
create pfile='/home/oracle/ORACLE_UNQNAME.pfile' from spfile;
exit

[oracle@]$  sqlplus "/ as sysdba"
SQL> @rm_dbfiles.sql
SQL> exit
[oracle@ demo]$ chmod 777 files.lst
[oracle@ demo]$ srvctl stop database -d <db_unique_name>

Save and Execute the Script
[oracle@sndel demo]$ ./files.lst
```

All files for the starter database have now been removed. In case of DB on IaaS, since the starter database will be initially created this step is not required.

The entry will look similar to the following:

```
[oracle@sndeldr ~]$ pwd
/home/oracle
[oracle@sndeldr ~]$ vi rm_dbfiles.sql
[oracle@sndeldr ~]$ chmod +x rm_dbfiles.sql


[oracle@sndeldr ~]$ ls -ltra

-rw-r--r--   1 oracle oinstall  172 Nov 23  2021 .kshrc
-rw-r--r--   1 oracle oinstall   18 Nov 23  2021 .bash_logout
-rw-r--r--   1 oracle oinstall  203 Mar  6 13:14 .bash_profile
-rw-r--r--   1 oracle oinstall  635 Mar  6 13:47 .bashrc
asmcmd rm +DATA/SNDEL_S/DATAFILE/system.261.1162906773
drwxr-xr-x   2 oracle oinstall 4096 Mar  6 13:47 .ssh
-rwxr-xr-x   1 oracle oinstall  340 Mar  7 13:12 rm_dbfiles.sql
-rw-------   1 oracle oinstall  112 Mar  7 13:13 .bash_history


[oracle@sndeldr ~]$ sqlplus "/ as sysdba"
SQL*Plus: Release 19.0.0.0.0 - Production on Thu Mar 7 13:15:53 2024
Version 19.20.0.0.0
Copyright (c) 1982, 2022, Oracle.  All rights reserved.
Connected to:
Oracle Database 19c EE Extreme Perf Release 19.0.0.0.0 - Production
Version 19.20.0.0.0

SQL> @rm_dbfiles.sql
asmcmd rm +DATA/SNDEL_S/DATAFILE/system.261.1162906773
asmcmd rm +DATA/SNDEL_S/DATAFILE/sysaux.268.1162906761
asmcmd rm +DATA/SNDEL_S/DATAFILE/undotbs1.260.1162906793
asmcmd rm
+DATA/SNDEL_S/02A979CA24559A30E0630706F40AB807/DATAFILE/system.264.1162906537
asmcmd rm
+DATA/SNDEL_S/02A979CA24559A30E0630706F40AB807/DATAFILE/sysaux.265.1162906537
asmcmd rm
+DATA/SNDEL_S/02A979CA24559A30E0630706F40AB807/DATAFILE/undotbs1.266.1162906537
asmcmd rm
+DATA/SNDEL_S/12FF285E05526CA0E063D700000A1259/DATAFILE/system.273.1162907089
```

```
asmcmd rm
+DATA/SNDEL_S/12FF285E05526CA0E063D700000A1259/DATAFILE/sysaux.271.1162907097
asmcmd rm
+DATA/SNDEL_S/12FF285E05526CA0E063D700000A1259/DATAFILE/undotbs1.270.1162907105
asmcmd rm +DATA/SNDEL_S/DATAFILE/users.269.1162907239
asmcmd rm
+DATA/SNDEL_S/12FF285E05526CA0E063D700000A1259/DATAFILE/users.274.1162907239
asmcmd rm +DATA/SNDEL_S/TEMPFILE/temp.263.1162906845
asmcmd rm
+DATA/SNDEL_S/12FF0A64F19E2F62E063D700000A3A33/TEMPFILE/temp.267.1162906565
asmcmd rm
+DATA/SNDEL_S/12FF285E05526CA0E063D700000A1259/TEMPFILE/temp.272.1162907085
asmcmd rm +RECO/SNDEL_S/ONLINELOG/group_3.259.1162906499
asmcmd rm +RECO/SNDEL_S/ONLINELOG/group_2.258.1162906499
asmcmd rm +RECO/SNDEL_S/ONLINELOG/group_1.257.1162906499
not spooling currently
Disconnected from Oracle Database 19c EE Extreme Perf Release 19.0.0.0.0 -
Production
Version 19.20.0.0.0

[oracle@sndeldr ~]$ vi files.lst
[oracle@sndeldr ~]$ chmod 777 files.lst

[oracle@sndeldr ~]$ srvctl stop database -d sndel_s
[oracle@sndeldr ~]$ ./files.lst
[oracle@sndeldr ~]$ asmcmd
ASMCMD> ls
ASMCMD> lsdg
ASMCMD> exit

[oracle@sndeldr ~]$
```

- Copy the Password File and wallet file

Copy the database password file and the wallet files received in /tmp from Primary to the respective locations on the Standby database DB system.

```
[opc@ tmp]$  sudo cp /tmp/orapw<sid> $ORACLE_HOME/dbs/orapw<sid>
```

The output will look similar to the following:

```
[opc@sndeldr ~]$ sudo chown oracle:oinstall /tmp/orapwsndel
[opc@sndeldr ~]$ sudo chown oracle:asmadmin /tmp/ewallet.p12
[opc@sndeldr ~]$ sudo chown oracle:asmadmin /tmp/cwallet.sso

[opc@sndel]$ sudo ls -ltra /u01/app/oracle/product/19.0.0.0/dbhome_1/dbs/orapwsndel
-rw-r-----1 oracle oinstall 2048 Jul3 13:42
/u01/app/oracle/product/19.0.0.0/dbhome_1/dbs/orapwsndel
```

- Verify that the md5sum output of password file matches on Primary and Standby

```
md5sum /u01/app/oracle/product/19.0.0.0/dbhome_1/dbs/orapw<sid>
```

The output will look similar to the following:

- Primary

```
[oracle@ ~]$  md5sum /u01/app/oracle/product/19.0.0.0/dbhome_1/dbs/orapwsndel
b3895fa6357471f80c6e0f4ac16fdc23
/u01/app/oracle/product/19.0.0.0/dbhome_1/dbs/orapwsndel
```

- Standby

```
[oracle@ ~]$ md5sum /u01/app/oracle/product/19.0.0.0/dbhome_1/dbs/orapwsndel
b3895fa6357471f80c6e0f4ac16fdc23
/u01/app/oracle/product/19.0.0.0/dbhome_1/dbs/orapwsndel
```

- Remove existing wallet files and copy wallet files from /tmp to tde location

```
[oracle@ ~]$ cd /opt/oracle/dcs/commonstore/wallets/<db_unique_name>/tde/
[oracle@tde]$ rm ewallet.p12 cwallet.sso

[opc@tmp]$ sudo cp /tmp/ewallet.p12
/opt/oracle/dcs/commonstore/wallets/<db_unique_name>/tde/ewallet.p12
[opc@tmp]$ sudo cp /tmp/cwallet.sso
/opt/oracle/dcs/commonstore/wallets/<db_unique_name>/tde/cwallet.sso

[opc@sndel ~]$ sudo chown oracle:asmadmin
/opt/oracle/dcs/commonstore/wallets/sndel_s/tde/ewallet.p12
[opc@sndel ~]$  sudo chown oracle:asmadmin
/opt/oracle/dcs/commonstore/wallets/sndel_s/tde/cwallet.sso
```

- Configure static listener

On the standby database in OCI, append the SID_LIST_LISTENER file to include the database unique name, Oracle Home of OCI, and the Oracle System Identifier (SID) of the primary database.

```
[grid@sndel]$ lsnrctl status
[grid@sndel]$ vi listener.ora

Add the following entry to listener.ora file-
SID_LIST_LISTENER=(SID_LIST=(SID_DESC=(GLOBAL_DBNAME=DB_UNIQUE_NAME of the
primary database)(ORACLE_HOME=Local Oracle Home of the primary database)
(SID_NAME = ORACLE SID of the primary database)))


[grid@sndeldr ~]$ vi /u01/app/19.0.0.0/grid/network/admin/listener.ora
[grid@sndeldr ~]$ lsnrctl reload
LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 08-MAR-2024 11:30:18
Copyright (c) 1991, 2023, Oracle.  All rights reserved.
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER)))
The command completed successfully.
```

The entry will look similar to the following:

```
[grid@sndeldr ~]$ lsnrctl status
LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 08-MAR-2024 11:30:24
Copyright (c) 1991, 2023, Oracle.  All rights reserved.
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 19.0.0.0.0 - Production
Start Date                07-MAR-2024 10:01:36
Uptime                    1 days 1 hr. 28 min. 48 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /u01/app/19.0.0.0/grid/network/admin/listener.ora
Listener Log File
/u01/app/grid/diag/tnslsnr/sndeldr/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))
Services Summary...
Service "sndeldr.sub06291309370.snvcn.oraclevcn.com" has 1 instance(s).
  Instance "sndel", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully.
```

- Ammend the below entries in listener.ora file

```
SID_LIST_LISTENER=(SID_LIST=(SID_DESC=(GLOBAL_DBNAME=
sndel_s.sub06291309370.snvcn.oraclevcn.com)(ORACLE_HOME=/u01/app/oracle/produ
ct/19.0.0.0/dbhome_1)(SID_NAME=sndel)))

[grid@sndel]$ lsnrctl reload
[grid@sndel]$ lsnrctl status
```

- Configure tns entries.

```
[oracle@sndel]$ cd $ORACLE_HOME/network/admin
[oracle@sndel]$ vi tnsnames.ora
```

The entry in tnsnames.ora should look similar to the following:

```
SNDEL_S =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = sndel)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = sndel_s.sub06291309370.snvcn.oraclevcn.com)
    )
  )

SNDEL_P =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = sndel)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = sndel_p.sub06291309370.snvcn.oraclevcn.com)
    )
  )
```

Deploy

Create a Standby Database Using RMAN

- Connect to the database and start the database in nomount mode

```
$ rman target /
RMAN> startup nomount;
```

- Restore the standby control file from the primary service.

```
RMAN> restore standby controlfile from service <db_unique_name>;
```

In this example, sndel_p is the primary database.

The output will look similar to the following:

```
RMAN> restore standby controlfile from service 'sndel_p';
Starting restore at 04-JUL-23
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=28 device type=DISK
channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: using network backup set from service DBUKS_898_LHR
channel ORA_DISK_1: restoring control file
channel ORA_DISK_1: restore complete, elapsed time: 00:00:04
output file name=+RECO/sndel_s/CONTROLFILE/current.256.1139953721
Finished restore at 04-JUL-23
```

- Mount the database.

```
RMAN> alter database mount;
```

The output will look similar to the following:

```
RMAN> alter database mount;
released channel: ORA_DISK_1
Statement processed
```

- As root on the OCI instance, change the permissions of the Oracle directory to open (chmod 777).

```
[root@sndel]$ cd /opt/
[root@sndel]$ ls -ltra
drwxr-xr-x  10 root root  4096 Jun 20 03:52 oracle

[root@sndel]$ chmod 777 oracle/

[root@sndel]$ ls -ltra
drwxrwxrwx  10 root root  4096 Jun 20 03:52 oracle
```

- Restore the database from the primary database (sndel_p).

```
RMAN> restore database from service <db_unique_name>;
```

or

```
RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE DISK;
  restore database from service 'sndel_p';
}
```

The output will look similar to the following:

```
RMAN> RUN
{
  ALLOCATE CHANNEL c1 DEVICE TYPE DISK;
  restore database from service 'sndel_p';
}

allocated channel: c1
channel c1: SID=326 device type=DISK

Starting restore at 08-MAR-24

channel c1: starting datafile backup set restore
channel c1: using network backup set from service sndel_p
channel c1: specifying datafile(s) to restore from backup set
channel c1: restoring datafile 00001 to
+DATA/SNDEL_P/DATAFILE/system.261.1162902099
channel c1: restore complete, elapsed time: 00:00:07
channel c1: starting datafile backup set restore
channel c1: using network backup set from service sndel_p
channel c1: specifying datafile(s) to restore from backup set
channel c1: restoring datafile 00003 to
+DATA/SNDEL_P/DATAFILE/sysaux.268.1162902087
channel c1: restore complete, elapsed time: 00:00:08
channel c1: starting datafile backup set restore
channel c1: using network backup set from service sndel_p
channel c1: specifying datafile(s) to restore from backup set
channel c1: restoring datafile 00004 to
+DATA/SNDEL_P/DATAFILE/undotbs1.260.1162902119
channel c1: restore complete, elapsed time: 00:00:01
channel c1: starting datafile backup set restore
channel c1: using network backup set from service sndel_p
channel c1: specifying datafile(s) to restore from backup set
channel c1: restoring datafile 00005 to
+DATA/SNDEL_P/02A979CA24559A30E0630706F40AB807/DATAFILE/system.264.11629018
61
channel c1: restore complete, elapsed time: 00:00:03
channel c1: starting datafile backup set restore
channel c1: using network backup set from service sndel_p
channel c1: specifying datafile(s) to restore from backup set
channel c1: restoring datafile 00006 to
+DATA/SNDEL_P/02A979CA24559A30E0630706F40AB807/DATAFILE/sysaux.265.11629018
61
channel c1: restore complete, elapsed time: 00:00:04
channel c1: starting datafile backup set restore
channel c1: using network backup set from service sndel_p
```

```
channel c1: specifying datafile(s) to restore from backup set
channel c1: restoring datafile 00007 to
+DATA/SNDEL_P/02A979CA24559A30E0630706F40AB807/DATAFILE/undotbs1.266.116290
1861
channel c1: restore complete, elapsed time: 00:00:01
channel c1: starting datafile backup set restore
channel c1: using network backup set from service sndel_p
channel c1: specifying datafile(s) to restore from backup set
channel c1: restoring datafile 00008 to
+DATA/SNDEL_P/12FE1204497C090FE0633000000AB26F/DATAFILE/system.273.11629024
21
channel c1: restore complete, elapsed time: 00:00:03
channel c1: starting datafile backup set restore
channel c1: using network backup set from service sndel_p
channel c1: specifying datafile(s) to restore from backup set
channel c1: restoring datafile 00009 to
+DATA/SNDEL_P/12FE1204497C090FE0633000000AB26F/DATAFILE/sysaux.271.11629024
29
channel c1: restore complete, elapsed time: 00:00:04
channel c1: starting datafile backup set restore
channel c1: using network backup set from service sndel_p
channel c1: specifying datafile(s) to restore from backup set
channel c1: restoring datafile 00010 to
+DATA/SNDEL_P/12FE1204497C090FE0633000000AB26F/DATAFILE/undotbs1.270.116290
2437
channel c1: restore complete, elapsed time: 00:00:01
channel c1: starting datafile backup set restore
channel c1: using network backup set from service sndel_p
channel c1: specifying datafile(s) to restore from backup set
channel c1: restoring datafile 00011 to
+DATA/SNDEL_P/DATAFILE/users.269.1162902567
channel c1: restore complete, elapsed time: 00:00:02
channel c1: starting datafile backup set restore
channel c1: using network backup set from service sndel_p
channel c1: specifying datafile(s) to restore from backup set
channel c1: restoring datafile 00012 to
+DATA/SNDEL_P/12FE1204497C090FE0633000000AB26F/DATAFILE/users.274.116290256
7
channel c1: restore complete, elapsed time: 00:00:01
Finished restore at 08-MAR-24
released channel: c1

RMAN> exit;
```

Standby Validation

- View the database name and role.

```
SQL> select FORCE_LOGGING, FLASHBACK_ON, OPEN_MODE, DATABASE_ROLE,
DATAGUARD_BROKER, PROTECTION_MODE from v$database ;

FORCE_LOGGING     FLASHBACK_ON      OPEN_MODE      DATABASE_ROLE     DATAGUAR
PROTECTION_MODE
------------------------------------------------------------------    ---------- ----------
YES               NO                MOUNTED        PHYSICAL STANDBY  DISABLED   MAXIMUM
PERFORMANCE
```

- View the archive process, status thread number and sequence number for the database.

```
SQL> select sysdate,process,status,thread#,sequence#,block#  from
v$managed_standby  where status!='IDLE';
04-JUL-23 ARCH      CONNECTED             0          0          0
04-JUL-23 DGRD      ALLOCATED             0          0          0
04-JUL-23 DGRD      ALLOCATED             0          0          0
04-JUL-23 ARCH      CONNECTED             0          0          0
04-JUL-23 ARCH      CONNECTED             0          0          0
04-JUL-23 ARCH      CONNECTED             0          0          0


SQL> select distinct process from gv$managed_standby;

PROCESS
---------
ARCH
DGRD
```

- Add Standby logfiles to primary and standby databases.

On the Primary database

```
select group#, type, member from v$logfile;
select bytes, group# from v$log;
alter database add standby logfile thread 1 group <number>('+RECO') size
<bytes>;
```

The output will look similar to the following:

```
SQL> select group#, type, member from v$logfile;

    GROUP# TYPE    MEMBER
---------- ------- ------------------------------------------------------------
         3 ONLINE  +RECO/SNDEL_P/ONLINELOG/group_3.259.1139942665
         2 ONLINE  +RECO/SNDEL_P/ONLINELOG/group_2.258.1139942665
         1 ONLINE  +RECO/SNDEL_P/ONLINELOG/group_1.257.1139942665
```

```
SQL> select bytes, group# from v$log;
1073741824          1
1073741824          3
1073741824          2

SQL> alter database add standby logfile thread 1 group 4('+RECO') size
1073741824;
Database altered.
SQL> alter database add standby logfile thread 1 group 5('+RECO') size
1073741824;
Database altered.
SQL> alter database add standby logfile thread 1 group 6('+RECO') size
1073741824;
Database altered.
SQL> alter database add standby logfile thread 1 group 7('+RECO') size
1073741824;
Database altered.


SQL> select group#, type, member from v$logfile;

    GROUP# TYPE    MEMBER
---------- ------- ----------------------------------------
       3 ONLINE  +RECO/SNDEL_P/ONLINELOG/group_3.259.1162901821
       2 ONLINE  +RECO/SNDEL_P/ONLINELOG/group_2.258.1162901821
       1 ONLINE  +RECO/SNDEL_P/ONLINELOG/group_1.257.1162901821
       4 STANDBY +RECO/SNDEL_P/ONLINELOG/group_4.265.1162908055
       5 STANDBY +RECO/SNDEL_P/ONLINELOG/group_5.266.1162908061
       6 STANDBY +RECO/SNDEL_P/ONLINELOG/group_6.267.1162908067
       7 STANDBY +RECO/SNDEL_P/ONLINELOG/group_7.268.1162908075
7 rows selected.
```

On the Standby database

```
select group#, type, member from v$logfile;
select bytes, group# from v$log;
alter database add standby logfile group <number>('+RECO') size <bytes>;
```

The output will look similar to the following:

```
SQL> select group#, type, member from v$logfile;

    GROUP# TYPE    MEMBER
---------- ------- ------------------------------------------------------------
         3 ONLINE  +RECO/SNDEL_S/ONLINELOG/group_3.259.1139942665
         2 ONLINE  +RECO/SNDEL_S/ONLINELOG/group_2.258.1139942665
         1 ONLINE  +RECO/SNDEL_S/ONLINELOG/group_1.257.1139942665

SQL> select bytes, group# from v$log;

     BYTES     GROUP#
---------- ----------
1073741824      1
1073741824      2
1073741824      3
```

```
SQL> alter database add standby logfile thread 1 group 4('+RECO') size
1073741824;
Database altered.
SQL> alter database add standby logfile thread 1 group 5('+RECO') size
1073741824;
Database altered.
SQL> alter database add standby logfile thread 1 group 6('+RECO') size
1073741824;
Database altered.
SQL> alter database add standby logfile thread 1 group 7('+RECO') size
1073741824;
Database altered.


SQL> select group#, type, member from v$logfile;

    GROUP# TYPE    MEMBER
---------- ------- ------------------------------------------
       3 ONLINE  +RECO/SNDEL_S/ONLINELOG/group_3.259.1162901821
       2 ONLINE  +RECO/SNDEL_S/ONLINELOG/group_2.258.1162901821
       1 ONLINE  +RECO/SNDEL_S/ONLINELOG/group_1.257.1162901821
       4 STANDBY +RECO/SNDEL_S/ONLINELOG/group_4.265.1162908055
       5 STANDBY +RECO/SNDEL_S/ONLINELOG/group_5.266.1162908061
       6 STANDBY +RECO/SNDEL_S/ONLINELOG/group_6.267.1162908067
       7 STANDBY +RECO/SNDEL_S/ONLINELOG/group_7.268.1162908075
7 rows selected.
```

Configure Oracle Data Guard Broker

Configure Oracle Data Guard, register the standby database, and synchronize primary and standby databases. Configure Oracle Data Guard by enabling the dg_broker_config_file parameter on the primary and standby databases. For Oracle Automatic Storage Management (Oracle ASM), place the broker configuration files on separate disk groups. For Oracle Real Application Clusters (Oracle RAC), broker configuration files must be on shared storage.

- Check the dg_broker_start value for the primary database and standby database. Ensure it is set to True.

```
SQL> show parameter dg_broker_start;
```

The output will look similar to the following:

```
NAME                 TYPE                 VALUE
-------------------  -----------------    -----------
dg_broker_start      boolean              TRUE
```

If the dg_broker_start is set to FALSE, then set the parameter to TRUE before proceeding to the next steps.

```
SQL> alter system set dg_broker_start=true;

SQL> select pname from v$process where pname like 'DMON%';
PNAME
-----
DMON
```

- Check the Oracle Data Guard files for the primary database.

```
SQL> show parameter dg_broker_config_file1;
dg_broker_config_file1             string  +DATA/SNDEL_P/dr1sndel_p.dat

SQL> show parameter dg_broker_config_file2;
dg_broker_config_file2             string  +DATA/SNDEL_P/dr2sndel_p.dat
```

If you have Oracle RAC or Oracle ASM, then you can change the configuration file location. Example-

```
SQL> alter system set dg_broker_config_file1=broker_config_file location;
SQL> alter system set dg_broker_config_file2=broker_config_file location;
```

- Register the Primary and Standby Database

Use the Oracle Data Guard command-line interface (DGMGRL) to register the primary database and to add the standby database profile to the broker configuration.

Log into the primary host as sys.

```
$ dgmgrl sys/<password>@<net service name for primary database>
```

- Create a configuration that uses the primary database name.

```
DGMGRL> CREATE CONFIGURATION <configuration_name> AS PRIMARY DATABASE IS
<primary database name> CONNECT IDENTIFIER IS <primary database name>;
```

The output will look similar to the following:

```
CREATE CONFIGURATION sndel_p_sndel_s AS PRIMARY DATABASE IS sndel_p CONNECT
IDENTIFIER IS sndel_p;
```

In case of OCI environment where there is already a Data Guard association enabled, then the above step is not required and only the additional manually created Data Guard can be added to the existing configuration.

- Add the standby database.

```
DGMGRL> ADD DATABASE <standby unique database name> AS CONNECT IDENTIFIER
IS <standby unique database name> MAINTAINED AS PHYSICAL;
```

The output will look similar to the following:

```
DGMGRL> ADD DATABASE sndel_s AS CONNECT IDENTIFIER IS sndel_s MAINTAINED AS
PHYSICAL;

Database "sndel_s" added
```

- Enable the configuration.

```
DGMGRL> enable configuration;
```

- Display the configuration.

```
DGMGRL> show configuration;
```

Note: If you receive a WARNING that the apply lag could not be determined, log in to the primary database through sqlplus and do some log switches.

The output will look similar to the following:

```
DGMGRL> show configuration;

Configuration - sndel_p_sndel_s
  Protection Mode: MaxPerformance

Members:
  sndel_p - Primary database
    sndel_s - Physical standby database

Fast-Start Failover:  Disabled
Configuration Status: SUCCESS (status updated 36 seconds ago)



DGMGRL> show database sndel_p
Database – sndel_p

Role:               PRIMARY
  Intended State:     TRANSPORT-ON
  Instance(s): sndel
Database Status: SUCCESS



DGMGRL> show database sndel_s
Database – sndel_s

Role:               PHYSICAL STANDBY
  Intended State:     APPLY-ON
  Transport Lag:      0 seconds (computed 1 second ago)
  Apply Lag:          0 seconds (computed 0 seconds ago)
  Average Apply Rate: 59.00 KByte/s
  Real Time Query:    ON
  Instance(s): sndel
Database Status: SUCCESS
```

You can set the state of the Physical standby database to apply-off or apply-on using the below commands in dgmgrl

```
edit database sndel_s set state=apply-off;

edit database sndel_s set state=apply-on;
```

Confirm the Primary and Standby replication

On the Standby database

```
select sysdate, process, status, thread#, sequence#, block#
from v$managed_standby
where status!='IDLE';
```

The output will look similar to the following:

```
SQL> select sysdate, process, status, thread#, sequence#, block#
from v$managed_standby
where status!='IDLE';
SYSDATE   PROCESS   STATUS          THREAD#  SEQUENCE#     BLOCK#
--------- --------- ------------ ---------- ---------- ----------
10-JUL-23 DGRD      ALLOCATED             0          0          0
10-JUL-23 ARCH      CONNECTED             0          0          0
10-JUL-23 DGRD      ALLOCATED             0          0          0
10-JUL-23 ARCH      CONNECTED             0          0          0
10-JUL-23 ARCH      CONNECTED             0          0          0
10-JUL-23 ARCH      CONNECTED             0          0          0
10-JUL-23 RFS       RECEIVING             1        417       2413
10-JUL-23 MRP0      APPLYING_LOG          1        417       2412
```

- View the Data guard processes.

```
SQL> select distinct process from gv$managed_standby;
PROCESS
---------
DGRD
RFS
MRP0
ARCH
```

- The Standby database has now started applying the log. Do some log switches on Primary database and confirm again on the standby database.

On the Primary database

```
SQL> archive log list
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     415
Next log sequence to archive   417
Current log sequence           417

SQL> alter system switch logfile;
System altered.
```

```
SQL> /
System altered.

SQL> archive log list
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     417
Next log sequence to archive   419
Current log sequence           419
```

On the Standby database

```
SQL> select sysdate, process, status, thread#, sequence#, block#
from v$managed_standby
where status!='IDLE';

SYSDATE   PROCESS   STATUS          THREAD#  SEQUENCE#     BLOCK#
--------- --------- ------------ ---------- ---------- ----------
10-JUL-23 DGRD      ALLOCATED             0          0          0
10-JUL-23 ARCH      CONNECTED             0          0          0
10-JUL-23 DGRD      ALLOCATED             0          0          0
10-JUL-23 ARCH      CLOSING               1        418          1
10-JUL-23 ARCH      CONNECTED             0          0          0
10-JUL-23 ARCH      CLOSING               1        417       2048
10-JUL-23 RFS       RECEIVING             1        419         59
10-JUL-23 MRP0      APPLYING_LOG          1        419         58
```

- Connect using dgmgrl on standby or primary and set apply to OFF

Login with dgmgrl and set the state of the Physical standby database to apply-off

```
DGMGRL> show database sndel_s
Database - sndel_s

Role:               PHYSICAL STANDBY
  Intended State:     APPLY-ON
  Transport Lag:      0 seconds (computed 1 second ago)
  Apply Lag:          0 seconds (computed 0 seconds ago)
  Average Apply Rate: 59.00 KByte/s
  Real Time Query:    ON
  Instance(s): sndel
Database Status: SUCCESS

DGMGRL> edit database sndel_s set state=apply-off;

DGMGRL> show database sndel_s
Database - sndel_s
  Role:               PHYSICAL STANDBY
  Intended State:     APPLY-OFF
  Transport Lag:      0 seconds (computed 1 second ago)
  Apply Lag:          0 seconds (computed 0 seconds ago)
  Average Apply Rate: 59.00 KByte/s
  Real Time Query:    ON
  Instance(s): sndel
Database Status: SUCCESS
```

- Enable flashback on the standby database

Using sqlplus

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE cancel;
Database altered.

SQL> alter database flashback on;
Database altered
```

- Confirm that database flashback is enabled (on).

```
SQL> select flashback_on from v$database;
```

The output will look similar to the following:

```
FLASHBACK_ON
------------------
YES
```

- Connect using dgmgrl on standby or primary and set apply to ON.

Login with dgmgrl and set the state of the Physical standby database to apply-on

```
DGMGRL> show database sndel_s
Database - sndel_s

Role:               PHYSICAL STANDBY
  Intended State:     APPLY-OFF
  Transport Lag:      0 seconds (computed 1 second ago)
  Apply Lag:          0 seconds (computed 0 seconds ago)
  Average Apply Rate: 59.00 KByte/s
  Real Time Query:    ON
  Instance(s): sndel
Database Status: SUCCESS

DGMGRL> edit database sndel_s set state=apply-on;

DGMGRL> show database sndel_s
Database - sndel_s
  Role:               PHYSICAL STANDBY
  Intended State:     APPLY-ON
  Transport Lag:      0 seconds (computed 1 second ago)
  Apply Lag:          0 seconds (computed 0 seconds ago)
  Average Apply Rate: 59.00 KByte/s
  Real Time Query:    ON
  Instance(s): sndel
Database Status: SUCCESS
```

- View the database details

```
SQL> select FORCE_LOGGING, FLASHBACK_ON, OPEN_MODE, DATABASE_ROLE,
DATAGUARD_BROKER, PROTECTION_MODE from v$database ;

FORCE_LOGGING   FLASHBACK_ON   OPEN_MODE    DATABASE_ROLE      DATAGUARD_BROKER     PROTECTION
_MODE
--------------  -------------- -----------  --------------     -----------------    ----------
YES             YES            MOUNTED       PHYSICAL STANDBY  ENABLED              MAXIMUM
PERFORMANCE
```

- Stop the standby database by using the Server Control (srvctl) utility.

Using srvctl utility to stop the database:

```
srvctl stop database -d <db_unique_name>
```

The output will look similar to the following:

```
srvctl stop database -d sndel_s
```

- Start the standby database by using the Server Control (srvctl) utility.

Using srvctl utility to start the database:

```
srvctl start database -d <db_unique_name>
```

The output will look similar to the following:

```
srvctl start database -d sndel_s
```

- View the database details *(Active Data Guard)*

```
SQL> select FORCE_LOGGING, FLASHBACK_ON, OPEN_MODE, DATABASE_ROLE,
DATAGUARD_BROKER, PROTECTION_MODE from v$database ;

FORCE_LOGGING  FLASHBACK_ON  OPEN_MODE             DATABASE_ROLE     DATAGUARD_BROKER
PROTECTION_MODE
-------------- ------------- --------------------  ----------------- ----------------  -------
YES            YES           READ ONLY WITH APPLY  PHYSICAL STANDBY   ENABLED      MAXIMUM
PERFORMANCE
```

- You may now give some more logfile switches from the primary and ensure that the redo is being applied to the standby.

Validate DR Readiness

Review best practices for disaster recovery readiness and verify your disaster recovery setup for planned and unplanned events.

Switch the Primary Database to Standby Database

When you have a planned activity, such as maintenance, you can make the current standby database the primary database with zero data loss. A switchover is a planned event that's initiated on the Primary database and completed on the standby database.

Log into a DGMGRL session on the Primary database with a sys username and password.

- On the Primary database, validate that the standby database is ready for switchover and failover.

In this example, the primary database is sndel_p and the standby database is sndel_s.

```
DGMGRL> validate database sndel_s (Standby Database)
```

The output will look similar to the following:

```
DGMGRL> validate database sndel_s
  Database Role:     Physical standby database
  Primary Database:  sndel_p
  Ready for Switchover:  Yes
  Ready for Failover:    Yes (Primary Running)
  Managed by Clusterware:
    sndel_p:  YES
    sndel_s:  YES
```

- On the primary database, execute the command to switchover to the standby database.

```
DGMGRL> switchover to sndel_s (Standby Database)
```

The output will look similar to the following:

```
DGMGRL> switchover to sndel_s
Performing switchover NOW, please wait...
Operation requires a connection to database "sndel_s"
Connecting ...
Connected to "sndel_s"
Connected as SYSDBA.
New primary database "sndel_s" is opening...
Oracle Clusterware is restarting database "sndel_p" ...
Connected to an idle instance.
Connected to "sndel_p"
Switchover succeeded, new primary is "sndel_s"

DGMGRL> validate database sndel_s

  Database Role:     Primary database
  Ready for Switchover:  Yes
  Managed by Clusterware:
    Sndel_s:  YES
```

```
DGMGRL> validate database sndel_p
  Database Role:     Physical standby database
  Primary Database:  sndel_s

  Ready for Switchover:  Yes
  Ready for Failover:    Yes (Primary Running)

  Managed by Clusterware:
    dbuks_r2j_ams:  YES
    dbuks_898_lhr:  YES
```

```
DGMGRL> show configuration;

Configuration - sndel_p_sndel_s
  Protection Mode: MaxPerformance
  Members:
  sndel_s - Primary database
    sndel_p - Physical standby database

Fast-Start Failover:  Disabled
Configuration Status:
SUCCESS (status updated 52 seconds ago)
```

- On the new remote primary database, determine the open mode status.

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

The output will look similar to the following:

```
OPEN_MODE
----------
READ WRITE
```

- On the previous primary database which is now the standby, determine the open mode status.

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

The output will look similar to the following:

```
OPEN_MODE
--------------------
READ ONLY WITH APPLY
```

The standby database is now the primary database.

Switch back to the Original Primary database

- Connect to dgmgrl from new Primary.

```
[oracle@sndel]$ dgmgrl
DGMGRL for Linux: Release 19.0.0.0.0 - Production on Mon Jul 10 06:59:41
2023
Version 19.19.0.0.0
Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights
reserved.
Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/<password>
Connected to "sndel_s"
Connected as SYSDBA.


DGMGRL> show configuration;
Configuration - sndel_p_sndel_s
  Protection Mode: MaxPerformance
  Members:
  sndel_s - Primary database
  sndel_p - Physical standby database
Fast-Start Failover:  Disabled
Configuration Status:
SUCCESS (status updated 39 seconds ago)


DGMGRL> validate database sndel_s;
  Database Role:    Primary database
  Ready for Switchover:  Yes
  Managed by Clusterware:
    sndel_s:  YES


DGMGRL> validate database sndel_p;
  Database Role:     Physical standby database
  Primary Database:  sndel_s
  Ready for Switchover:  Yes
  Ready for Failover:    Yes (Primary Running)
  Managed by Clusterware:
    sndel_s:  YES
    sndel_p:  YES


DGMGRL> switchover to sndel_p ;
Performing switchover NOW, please wait...
Operation requires a connection to database "sndel_p"
Connecting ...
Connected to "sndel_p"
Connected as SYSDBA.
New primary database "sndel_p" is opening...
Oracle Clusterware is restarting database "sndel_s" ...
Connected to an idle instance.
Connected to an idle instance.
Connected to "dbuks_r2j_ams"
Switchover succeeded, new primary is "sndel_p"
```

The switchover is now completed.