

How to build Oracle 19c R2 standby database using Data Guard?

Version-1.2 | Last updated Date: 07/24/2020

Table of Contents

Task:	2
Part-1: Technical Steps On Primary Database Server:	2
Part-2: Technical Steps On Standby Database Server:	12
Part-3: Open up PRIMARY and STANDBY database ALERT LOG	20
Part-4: Setup Data Broker	21
Part-5: Changing Data Guard Protection Modes	29
Part-6: Data Broker Troubleshooting Commands	34
Part-7: Errors and troubleshooting	34

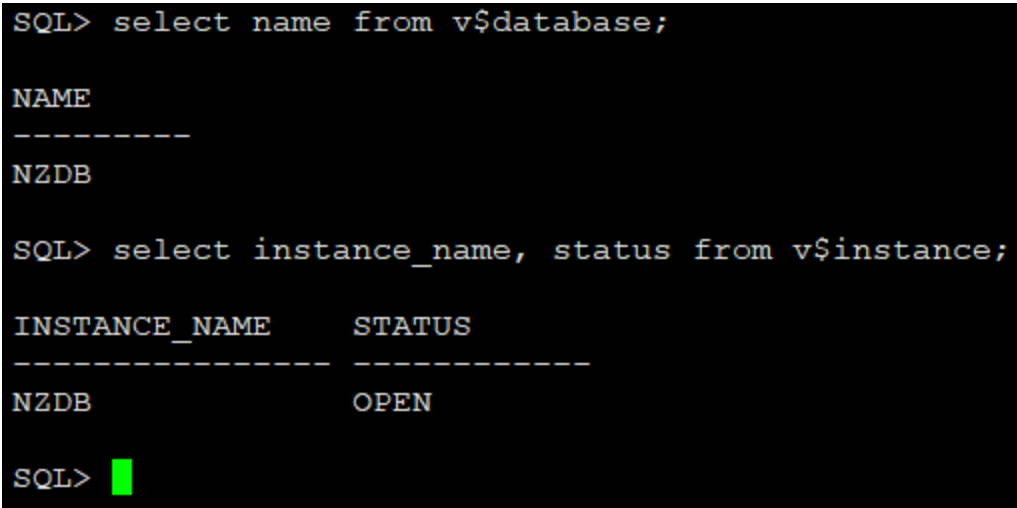
Task:

The task is to create a database called MOIDDGDB in the primary data center (in FL) and then build a standby database in the Disaster Recovery Data Center (in New York). Below you can find more details:

	Primary	Standby
Data Center Location	FL	New York
Hostname	Server-243	Server-233
Short Hostname	FL	ny
IP Address	192.168.0.223	192.168.0.233
Listener Name	LISTENER	LISTENER
Listener Port	1521	1521
Database Name	MOIDDGDB	MOIDDGDB
DB Unique Name	MOIDDGDB_FL	MOIDDGDB_NY

Part-1: Technical Steps On Primary Database Server:

No.	Description	Command / Details / Screenshots
1.1	Connect to Primary Database Server	- Putty to Primary Database Server - su - oracle
1.2	Create a new database called MOIDDGDB	- Create a new database called MOIDDGDB on the source server. How to create 19exic R3 database using scripts? -Moid
1.3	Create a script called db role to check the role of the database	<pre>vi /tmp/dbrole.sql i → to go insert mode</pre>

		<pre> set linesize 200 col open_mode format a12 col db_unique_name format a14 select name, OPEN_MODE, LOG_MODE, PROTECTION_MODE, DATABASE_ROLE from v\$database; </pre> ESC :wq!
1.4	Verify Primary database is OPEN	• . oraenv → MOIDDGDB • sqlplus / as sysdba ◦ select name from v\$database; <pre> ■ MOIDDGDB </pre> ◦ select instance_name from v\$instance; <pre> ■ MOIDDGDB </pre> ◦ select status from v\$instance; <pre> ■ OPEN </pre>  <pre> SQL> select name from v\$database; NAME ----- NZDB SQL> select instance_name, status from v\$instance; INSTANCE_NAME STATUS ----- NZDB OPEN SQL> █ </pre>
1.5	Verify Primary Server and to be Standby Server are in-network and reachable from PRIMARY Server.	• grep -iE 'FL NY' /etc/hosts • ping -c1 FL • ping -c1 NY

		<pre>Chi:(NZDB) Chicago \$ ping -c1 chi PING Chi (192.168.0.223) 56(84) bytes of data. 64 bytes from Chi (192.168.0.223): icmp_seq=1 ttl=64 time=0.055 ms --- Chi ping statistics --- 1 packets transmitted, 1 received, 0% packet loss, time 0ms rtt min/avg/max/mdev = 0.055/0.055/0.055/0.000 ms Chi:(NZDB) Chicago \$ ping -c1 NY PING NY (192.168.0.233) 56(84) bytes of data. 64 bytes from NY (192.168.0.233): icmp_seq=1 ttl=64 time=0.301 ms --- NY ping statistics --- 1 packets transmitted, 1 received, 0% packet loss, time 0ms rtt min/avg/max/mdev = 0.301/0.301/0.301/0.000 ms Chi:(NZDB) Chicago \$ █</pre>
1.6	Verify Database is in Archivelog mode.	• Archive LOG LIST; (OR) • select log_mode from v\$database; <pre>SQL> select log_mode from v\$database; LOG_MODE ----- ARCHIVELOG SQL> █</pre>
1.7	Enable Force Logging. This is to ensure changes are logged into redo logs.	• ALTER DATABASE FORCE LOGGING; <pre>SQL> ALTER DATABASE FORCE LOGGING; Database altered.</pre>

1.8	Verify force logging is enabled.	Select force_logging from v\$database; <pre>SQL> Select force_logging from v\$database; FORCE_LOGGING ----- YES</pre>
1.9	Set the standby file management to auto. This will ensure all datafiles get to create on standby when they are created on primary.	Alter system set standby_file_management=Auto scope=spfile; <pre>SQL> Alter system set standby_file_management=Auto scope=spfile; System altered.</pre>
1.10	Change db unique name.	Alter system set db_unique_name=MOIDDGDB_FL scope=spfile; <pre>SQL> Alter system set db_unique_name=NZDB_CHI scope=spfile; System altered.</pre>
1.11	Add standby redo logs	- ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 Group 4 ('/u15/oradata/MOIDDGDB/redo04a.rdo', '/u16/oradata/MOIDDGDB/redo04b.rdo', '/u17/oradata/MOIDDGDB/redo04c.rdo') size 20M; - ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 Group 5 ('/u15/oradata/MOIDDGDB/redo05a.rdo', '/u16/oradata/MOIDDGDB/redo05b.rdo', '/u17/oradata/MOIDDGDB/redo05c.rdo') size 20M; - ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 Group 6 ('/u15/oradata/MOIDDGDB/redo06a.rdo', '/u16/oradata/MOIDDGDB/redo06b.rdo', '/u17/oradata/MOIDDGDB/redo06c.rdo') size 20M; - ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 Group 7 ('/u15/oradata/MOIDDGDB/redo07a.rdo', '/u16/oradata/MOIDDGDB/redo07b.rdo', '/u17/oradata/MOIDDGDB/redo07c.rdo') size 20M;

	<pre>SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 Group 4 ('/u15/oradata/NZDB/redo04a.rdo', '/u16/oradata/NZDB/redo04b.rdo', '/u17/oradata/NZDB/redo04c.rdo') size 20M; ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 Group 5 ('/u15/oradata/NZDB/redo05a.rdo', '/u16/oradata/NZDB/redo05b.rdo', '/u17/oradata/NZDB/redo05c.rdo') size 20M; ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 Group 6 ('/u15/oradata/NZDB/redo06a.rdo', '/u16/oradata/NZDB/redo06b.rdo', '/u17/oradata/NZDB/redo06c.rdo') size 20M; ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 Group 7 ('/u15/oradata/NZDB/redo07a.rdo', '/u16/oradata/NZDB/redo07b.rdo', '/u17/oradata/NZDB/redo07c.rdo') size 20M; Database altered. SQL> Database altered.</pre>	
1.12	Verify all redo and log files are created.	<pre>set linesize 200; set pagesize 200; col member format a40; break on type skip 1; Select group#, type, member from v\$logfile order by 2,1;</pre>

		<pre> GROUP# TYPE MEMBER ----- 1 ONLINE /u16/oradata/NZDB/redo01b.rdo 1 /u15/oradata/NZDB/redo01a.rdo 1 /u17/oradata/NZDB/redo01c.rdo 2 /u15/oradata/NZDB/redo02a.rdo 2 /u16/oradata/NZDB/redo02b.rdo 2 /u17/oradata/NZDB/redo02c.rdo 3 /u16/oradata/NZDB/redo03b.rdo 3 /u15/oradata/NZDB/redo03a.rdo 3 /u17/oradata/NZDB/redo03c.rdo 4 STANDBY /u15/oradata/NZDB/redo04a.rdo 4 /u16/oradata/NZDB/redo04b.rdo 4 /u17/oradata/NZDB/redo04c.rdo 5 /u15/oradata/NZDB/redo05a.rdo 5 /u16/oradata/NZDB/redo05b.rdo 5 /u17/oradata/NZDB/redo05c.rdo 6 /u15/oradata/NZDB/redo06a.rdo 6 /u16/oradata/NZDB/redo06b.rdo 6 /u17/oradata/NZDB/redo06c.rdo 7 /u15/oradata/NZDB/redo07a.rdo 7 /u16/oradata/NZDB/redo07b.rdo 7 /u17/oradata/NZDB/redo07c.rdo 21 rows selected. </pre>
1.13	Shutdown database and exit out of SQL.	• shutdown immediate; • Exit; <pre> SQL> shutdown immediate; exit; Database closed. Database dismounted. ORACLE instance shut down. SQL> Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production Version 19.3.0.0.0 </pre>
1.14	Add DB_UNIQUE_NAME entry in	<pre>cat >> /etc/oratab << EOF</pre>

	oratab	MOIDDGDB_FL:/u01/app/oracle/product/19.3.0/dbhome_1:N EOF
1.15	Copy spfile to new instance spfile.	• <code>cd \$ORACLE_HOME/dbs</code> • <code>cp spfileMOIDDGDB.ora spfileMOIDDGDB_FL.ora</code>
1.16	Startup instance using db_unique_name	• <code>. oraenv --> MOIDDGDB_FL</code> • <code>sqlplus / as sysdba</code> • <code>startup</code> • <code>exit;</code> <pre>SQL> startup; ORACLE instance started. Total System Global Area 1073737800 bytes Fixed Size 8904776 bytes Variable Size 289406976 bytes Database Buffers 767557632 bytes Redo Buffers 7868416 bytes Database mounted. Database opened. SQL> exit; Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production Version 19.3.0.0.0 Chi: (NZDB_CHI) Chicago</pre>
1.17	Create password file	• <code>orapwd file=\$ORACLE_HOME/dbs/orapw\$ORACLE_SID password=abc123 format=12</code>
1.18	Create a directory for database backup.	• <code>mkdir -p /u99/MOIDDGDB/Backup</code>
1.19	Configure RMAN for backups	• <code>rman target /</code> ◦ <code>CONFIGURE RETENTION POLICY TO REDUNDANCY 1;</code> ◦ <code>CONFIGURE BACKUP OPTIMIZATION ON;</code> ◦ <code>CONFIGURE DEFAULT DEVICE TYPE TO DISK;</code> ◦ <code>CONFIGURE CONTROLFILE AUTOBACKUP ON;</code> ◦ <code>CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '/u99/MOIDDGDB/Backup/control_n_spfile_%F';</code>

		○ CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COMPRESSED BACKUPSET PARALLELISM 1; ○ CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/u99/MOIDDGDB/Backup/full_%d_%T_%t_%U.bak' MAXPIECESIZE 1 G; <pre> RMAN> new RMAN configuration parameters: CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '/u99/NZDB/Backup/control_n_spfile_%F' new RMAN configuration parameters are successfully stored RMAN> new RMAN configuration parameters: CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COMPRESSED BACKUPSET PARALLELISM 1; new RMAN configuration parameters are successfully stored RMAN> new RMAN configuration parameters: CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/u99/NZDB/Backup/full_%d_%T_%t_%U.bak' MAXPIECESIZE 1 G; new RMAN configuration parameters are successfully stored </pre> ● exit;
1.20	Add connection entries for Primary and Standby database in tnsnames.ora file.	<pre> cd \$ORACLE_HOME/network/admin cat >> tnsnames.ora << EOF MOIDDGDB_FL = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = FL) (PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SID = MOIDDGDB_FL))) MOIDDGDB_NY = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = NY) (PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) </pre>

		<pre> (SID = MOIDDGDB_NY))) EOF Chi:(NZDB_CHI) Chicago \$ cd \$ORACLE_HOME/network/admin Chi:(NZDB_CHI) Chicago \$ cat >> tnsnames.ora << EOF > NZDB_CHI = > (DESCRIPTION = > (ADDRESS = (PROTOCOL = TCP) (HOST = chi) (PORT = 1521)) > (CONNECT_DATA = > (SERVER = DEDICATED) > (SID = NZDB_CHI) >) >) > > NZDB_NY = > (DESCRIPTION = > (ADDRESS = (PROTOCOL = TCP) (HOST = NY) (PORT = 1521)) > (CONNECT_DATA = > (SERVER = DEDICATED) > (SID = NZDB_NY) >) >) > > EOF </pre>
1.21	Verify both connection strings are pingable	• <code>tnsping MOIDDGDB_FL</code> • <code>tnsping MOIDDGDB_NY</code>

		<pre> \$ tnsping NZDB_CHI TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 23-JUL-2020 23:16:19 Copyright (c) 1997, 2019, Oracle. All rights reserved. Used parameter files: Used TNSNAMES adapter to resolve the alias Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = chi)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SID = NZDB_CHI))) OK (0 msec) Chi: (NZDB_CHI) Chicago \$ tnsping NZDB_NY TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 23-JUL-2020 23:16:19 Copyright (c) 1997, 2019, Oracle. All rights reserved. Used parameter files: Used TNSNAMES adapter to resolve the alias Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = NY)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SID = NZDB_NY))) OK (0 msec) </pre>
1.22	Backup listener.ora file	<code>cp listener.ora listener.ora.bak</code>
1.23	Add the SID_DESC for primary database. Add another SID_DESC for data broker. <u>Note:</u> Add only the RED and GREEN part in listener.ora file from →	<pre> vi listener.ora LISTENER = (DESCRIPTION_LIST = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = FL)(PORT = 1521)) (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521)))) ADR_BASE_LISTENER = /u01/app/oracle SID_LIST_LISTENER = (SID_LIST = (SID_DESC = (GLOBAL_DBNAME = MOIDDGDB_FL) (ORACLE_HOME = /u01/app/oracle/product/19.3.0/dbhome_1) (SID_NAME = MOIDDGDB_FL)) (SID_DESC = (GLOBAL_DBNAME = MOIDDGDB_FL_DGMGRL) </pre>

		<pre>(SID_NAME = MOIDDGDB_FL) (ORACLE_HOME = /u01/app/oracle/product/19.3.0/dbhome_1)))</pre>
1.24	Reload Listener	- lsnrctl status listener grep \$ORACLE_SID - lsnrctl reload listener <pre>\$ lsnrctl status listener grep \$ORACLE_SID Instance "NZDB_CHI", status READY, has 1 handler(s) for this service... Service "NZDB_CHI" has 1 instance(s). Instance "NZDB_CHI", status READY, has 1 handler(s) for this service... Chi:(NZDB_CHI) Chicago \$ lsnrctl reload listener LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 23-JUL-2020 23:20:12 Copyright (c) 1991, 2019, Oracle. All rights reserved. Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=Chi) (PORT=1521))) The command completed successfully</pre>
1.25	Verify if the static services are registered in the listener	lsnrctl status listener grep \$ORACLE_SID grep READY <pre>\$ lsnrctl status listener grep \$ORACLE_SID grep READY Instance "NZDB_CHI", status READY, has 1 handler(s) for this service... Instance "NZDB_CHI", status READY, has 1 handler(s) for this service...</pre>

Part-2: Technical Steps On Standby Database Server:

No.	Description	Command / Details / Screenshots
2.1	Connect to Standby Database Server	- Putty to Standby Database Server - ssh oracle@ny -p23322

2.2	Verify Primary and Standby are in network and reachable from STANDBY Database Server.	• <code>grep -iE 'FL NY' /etc/hosts</code> • <code>ping -c1 FL</code> • <code>ping -c1 NY</code> <pre> NY:(NZDB) NY \$ grep -iE 'chi NY' /etc/hosts 192.168.0.223 Chi Chicago Linux-223 Server-223 192.168.0.233 NY NewYork Linux-233 Server-233 NY:(NZDB) NY \$ ping -c1 chi PING Chi (192.168.0.223) 56(84) bytes of data. 64 bytes from Chi (192.168.0.223): icmp_seq=1 ttl=64 time=0.217 ms --- Chi ping statistics --- 1 packets transmitted, 1 received, 0% packet loss, time 0ms rtt min/avg/max/mdev = 0.217/0.217/0.217/0.000 ms NY:(NZDB) NY \$ ping -c1 NY PING NY (192.168.0.233) 56(84) bytes of data. 64 bytes from NY (192.168.0.233): icmp_seq=1 ttl=64 time=0.057 ms --- NY ping statistics --- 1 packets transmitted, 1 received, 0% packet loss, time 0ms rtt min/avg/max/mdev = 0.057/0.057/0.057/0.000 ms NY:(NZDB) NY </pre>
2.3	Add db_name and db_unique_name entries in oratab file.	<pre> cat >> /etc/oratab << EOF MOIDDGDB:/u01/app/oracle/product/19c/dbhome_1:N MOIDDGDB_NY:/u01/app/oracle/product/19c/dbhome_1:N EOF </pre>
2.4	Create directories for database related files.	• <code>. oraenv → MOIDDGDB</code> ○ #Create Dump Directories: ■ <code>mkdir -p \$ORACLE_BASE/admin/\$ORACLE_SID/adump</code> ■ <code>mkdir -p \$ORACLE_BASE/admin/\$ORACLE_SID/bdump</code> ■ <code>mkdir -p \$ORACLE_BASE/admin/\$ORACLE_SID/cdump</code> ■ <code>mkdir -p \$ORACLE_BASE/admin/\$ORACLE_SID/udump</code> ○ #Create Controls file Directories: ■ <code>mkdir -p /u02/oradata/\$ORACLE_SID</code>

		■ mkdir -p /u03/oradata/\$ORACLE_SID ■ mkdir -p /u04/oradata/\$ORACLE_SID ○ #Create FRA ■ mkdir -p /u05/fast_recovery_area/\$ORACLE_SID ○ #Datafile Locations: ■ mkdir -p /u14/oradata/\$ORACLE_SID ○ #Redo Log Locations: ■ mkdir -p /u15/oradata/\$ORACLE_SID ■ mkdir -p /u16/oradata/\$ORACLE_SID ■ mkdir -p /u17/oradata/\$ORACLE_SID ○ #Redo Log Locations for standby Role. ■ mkdir -p /u15/oradata/\$ORACLE_SID ■ mkdir -p /u16/oradata/\$ORACLE_SID ■ mkdir -p /u17/oradata/\$ORACLE_SID ○ #Archive Log Location ■ mkdir -p /u98/\$ORACLE_SID/Archive ■ mkdir -p /u99/\$ORACLE_SID/Archive → for older docs. ○ #RMAN backup location ■ mkdir -p /u99/MOIDDGDB/Backup
2.5	Set oracle environment for standby database	● . oraenv → MOIDDGDB_NY
2.6	Add connection entries for Primary and Standby database in tnsnames.ora file.	<pre>cd \$ORACLE_HOME/network/admin cat >> tnsnames.ora << EOF MOIDDGDB_FL = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = FL) (PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SID = MOIDDGDB_FL)))</pre>

		<pre> MOIDDGDB_NY = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = NY) (PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SID = MOIDDGDB_NY))) EOF </pre> <pre> NY: (NZDB_NY) NY \$ cat >> tnsnames.ora << EOF > > NZDB_CHI = > (DESCRIPTION = > (ADDRESS = (PROTOCOL = TCP) (HOST = chi) (PORT = 1521)) > (CONNECT_DATA = > (SERVER = DEDICATED) > (SID = NZDB_CHI) >) >) > > NZDB_NY = > (DESCRIPTION = > (ADDRESS = (PROTOCOL = TCP) (HOST = NY) (PORT = 1521)) > (CONNECT_DATA = > (SERVER = DEDICATED) > (SID = NZDB_NY) >) >) > > EOF </pre>
2.7	Verify both connection strings are	• <code>tnsping MOIDDGDB_FL</code>

	pingable	<code>tnsping MOIDDGDB_NY</code> <pre> NY:(NZDB_NY) NY \$ tnsping NZDB_CHI TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 23-JUL-2020 23:07:52 Copyright (c) 1997, 2019, Oracle. All rights reserved. Used parameter files: Used TNSNAMES adapter to resolve the alias Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = chi) (PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SID = NZDB_CHI))) OK (0 msec) NY:(NZDB_NY) NY \$ tnsping NZDB_NY TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 23-JUL-2020 23:07:56 Copyright (c) 1997, 2019, Oracle. All rights reserved. Used parameter files: Used TNSNAMES adapter to resolve the alias Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = NY) (PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SID = NZDB_NY))) OK (0 msec) </pre>
2.8	Backup listener.ora file	<code>cp listener.ora listener.ora.bak</code>
2.9	Add the SID_DESC for primary database. Add another SID_DESC for data broker. Note: Add only the RED and GREEN part in listener.ora file from →	<code>vi listener.ora</code> <pre> LISTENER = (DESCRIPTION_LIST = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = ny) (PORT = 1521)) (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1521)))) ADR_BASE_LISTENER = /u01/app/oracle SID_LIST_LISTENER = (SID_LIST = (SID_DESC = (GLOBAL_DBNAME = MOIDDGDB_NY) (ORACLE_HOME = /u01/app/oracle/product/19.3.0/dbhome_1) (SID_NAME = MOIDDGDB_NY))) </pre>

		<pre>(SID_DESC = (GLOBAL_DBNAME = MOIDDGDB_NY_DGMGRL) (SID_NAME = MOIDDGDB_NY) (ORACLE_HOME = /u01/app/oracle/product/19.3.0/dbhome_1)))</pre>
2.10	Reload Listener	- lsnrctl status listener grep \$ORACLE_SID grep UNKNOWN - lsnrctl reload listener <pre>NY: (NZDB_NY) NY \$ lsnrctl status listener grep \$ORACLE_SID grep UNKNOWN NY: (NZDB_NY) NY \$ lsnrctl reload listener LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 23-JUL-2020 23:15:47 Copyright (c) 1991, 2019, Oracle. All rights reserved. Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=ny) (PORT=1521))) The command completed successfully</pre>
2.11	Verify if the static services are registered in listener	lsnrctl status listener grep \$ORACLE_SID grep UNKNOWN <pre>NY: (NZDB_NY) NY \$ lsnrctl status listener grep \$ORACLE_SID grep UNKNOWN Instance "NZDB_NY", status UNKNOWN, has 1 handler(s) for this service... Instance "NZDB_NY", status UNKNOWN, has 1 handler(s) for this service...</pre>
2.12	Copy password file from PRIMARY to STANDBY	scp -P24322 FL:\$ORACLE_HOME/dbs/orapwMOIDDGDB_FL \$ORACLE_HOME/dbs/orapwMOIDDGDB_NY
2.13	Copy dbrole.sql from PRIMARY	scp -P24322 FL:/tmp/dbrole.sql /tmp/.

2.14	Create a temporary init.ora file to start STANDBY database	• <code>vi /tmp/init\$ORACLE_SID.ora</code> ◦ i → to go to insert mode and add the only line: <pre>*.db_name='MOIDDGDB'</pre> ◦ ESC ◦ :wq!
2.15	Verify temporary init.ora is created	<code>ls -ltrh /tmp/init\$ORACLE_SID.ora</code>
2.16	Startup the standby instance	• <code>cd /tmp</code> • <code>sqlplus / as sysdba</code> • <code>startup nomount pfile='/tmp/initMOIDDGDB_NY.ora'</code> • <code>exit ;</code> <pre>SQL> startup nomount pfile='/tmp/initNZDB.ora'; ORACLE instance started. Total System Global Area 381677696 bytes Fixed Size 8896640 bytes Variable Size 314572800 bytes Database Buffers 50331648 bytes Redo Buffers 7876608 bytes</pre>
2.17	Using RMAN, connect to target (Primary) and auxiliary (standby) database. ~This is done in Target Database~	• <code>rman TARGET sys/abc123@MOIDDGDB_FL AUXILIARY sys/abc123@MOIDDGDB_NY</code> <pre>NY: (NZDB) NY \$ rman TARGET sys/abc123@NZDB_CHI AUXILIARY sys/abc123@NZDB_NY Recovery Manager: Release 19.0.0.0.0 - Production on Fri Jul 24 14:22:10 2020 Version 19.3.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. connected to target database: NZDB (DBID=3072663397) connected to auxiliary database (not started) RMAN></pre>

2.18	Start duplicating database from RMAN.	<pre> DUPLICATE TARGET DATABASE FOR STANDBY FROM ACTIVE DATABASE SPFILE SET db_unique_name='MOIDDGDB_NY' NOFILENAMECHECK; { switch clone datafile all; } executing Memory Script datafile 1 switched to datafile copy input datafile copy RECID=1 STAMP=1046620489 file name=/u14/oradata/NZDB/system01.dbf datafile 3 switched to datafile copy input datafile copy RECID=2 STAMP=1046620489 file name=/u14/oradata/NZDB/sysaux01.dbf datafile 4 switched to datafile copy input datafile copy RECID=3 STAMP=1046620490 file name=/u14/oradata/NZDB/undotbs01.dbf datafile 7 switched to datafile copy input datafile copy RECID=4 STAMP=1046620490 file name=/u14/oradata/NZDB/users01.dbf Finished Duplicate Db at 24-JUL-20 RMAN> █ </pre> <pre>exit;</pre>
2.19	Verify STANDBY database is created and in mount mode while database role is set to "Physical Standby"	• <code>sqlplus / as sysdba</code> • <code>@/tmp/dbrole.sql</code> <pre> SQL> @/tmp/dbrole.sql NAME OPEN_MODE LOG_MODE PROTECTION_MODE DATABASE_ROLE ----- NZDB MOUNTED ARCHIVELOG MAXIMUM PERFORMANCE PHYSICAL STANDBY SQL> █ </pre>

Part-3: Open up PRIMARY and STANDBY database ALERT LOG

3.1	Now open up two putty sessions to keep an eye on both alert logs. Once this is done, we will be switFLng the database role where standby will become primary and primary will become standby	On FL Server: • <code>cd /u01/app/oracle/diag/rdbms/MOIDDGDB_FL/MOIDDGDB_FL/trace</code> • <code>tail -f alert_MOIDDGDB_FL.ora</code> On NY Server: • <code>cd /u01/app/oracle/diag/rdbms/MOIDDGDB_NY/MOIDDGDB_NY/trace</code> • <code>tail -f alert_MOIDDGDB_NY.ora</code>
3.2	Check the status of database_role on primary and standby databases.	On FL Server: • <code>. oraenv → MOIDDGDB_FL</code> • <code>sqlplus / as sysdba</code> • <code>@/tmp/dbrole.sql</code> <pre>SQL> @/tmp/dbrole.sql NAME OPEN_MODE LOG_MODE PROTECTION_MODE DATABASE_ROLE ----- NZDB READ WRITE ARCHIVELOG MAXIMUM PERFORMANCE PRIMARY SQL> !hostname Chi</pre> On NY Server: • <code>. oraenv → MOIDDGDB_NY</code> • <code>sqlplus / as sysdba</code> • <code>@/tmp/dbrole.sql</code>

		<pre>SQL> @/tmp/dbrole.sql NAME OPEN_MODE LOG_MODE PROTECTION_MODE DATABASE_ROLE ----- NZDB MOUNTED ARCHIVELOG MAXIMUM PERFORMANCE PHYSICAL STANDBY SQL> !hostname NY SQL> █</pre>
--	--	---

Part-4: Setup Data Broker

No.	Description	Command / Details / Screenshots
4.1	Start Data Broker on Primary	• Putty to PRIMARY Database server • su - oracle • . oraenv → MOIDDGDB_FL • sqlplus / as sysdba ◦ ALTER SYSTEM SET dg_broker_start=true; ◦ show parameter dg_broker_start <pre>SQL> ALTER SYSTEM SET dg_broker_start=true; System altered. SQL> show parameter dg_broker_start NAME TYPE VALUE ----- dg_broker_start boolean TRUE SQL> █</pre>
4.2	Start Data Broker on Standby	• ssh oracle@ny -P24322 • . oraenv → MOIDDGDB_NY ◦ sqlplus / as sysdba ◦ ALTER SYSTEM SET dg_broker_start=true;

		<pre> o show parameter dg_broker_start SQL> ALTER SYSTEM SET dg_broker_start=true; System altered. SQL> show parameter dg_broker_start NAME TYPE VALUE ----- dg_broker_start boolean TRUE SQL> </pre>
4.3	Connect to data broker.	• Putty to PRIMARY Database server • su - oracle • dgmgrl sys/abc123@MOIDDGDB_FL <pre> Chi:(NZDB_CHI) Chicago \$ dgmgrl sys/abc123@NZDB_CHI DGMGRL for Linux: Release 19.0.0.0.0 - Production on Fri Jul 24 16:24:24 2020 Version 19.3.0.0.0 Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved. Welcome to DGMGRL, type "help" for information. Connected to "NZDB_CHI" Connected as SYSDBA. DGMGRL> </pre>
4.4	Create configuration to add primary.	<pre> CREATE CONFIGURATION MOIDDGDB_configuration AS PRIMARY DATABASE IS MOIDDGDB_FL CONNECT IDENTIFIER IS MOIDDGDB_FL; DGMGRL> CREATE CONFIGURATION NZDB_configuration AS PRIMARY DATABASE IS NZDB_CHI CONNECT IDENTIFIER IS NZDB_CHI; > > Configuration "nzdb_configuration" created with primary database "nzdb_chi" DGMGRL> </pre>
4.5	Add standby to broker configuration	<pre> ADD DATABASE MOIDDGDB_NY AS CONNECT IDENTIFIER IS MOIDDGDB_NY MAINTAINED AS PHYSICAL; </pre>

		<pre>DGMGRL> ADD DATABASE NZDB_NY AS CONNECT IDENTIFIER IS NZDB_NY MAINTAINED AS PHYSICAL; Database "nzdb_ny" added DGMGRL> █</pre>
4.6	Enable configuration	<pre>enable configuration; DGMGRL> enable configuration; Enabled. DGMGRL> █</pre>
4.7	Exit out of broker to perform few log switches.	• <code>exit;</code>
4.8	Perform a log switch	• <code>sqlplus / as sysdba</code> • <code>Alter system switch logfile;</code> <pre>SQL> Alter system switch logfile; System altered.</pre> • <code>exit;</code>
4.9	Login to broker and check the configuration	• <code>dgmgrl sys/abc123@MOIDDGDB_FL</code> • <code>show configuration;</code>

		<pre>DGMGRL> show configuration; Configuration - nzdb_configuration Protection Mode: MaxPerformance Members: nzdb_chi - Primary database nzdb_ny - Physical standby database Fast-Start Failover: Disabled Configuration Status: SUCCESS (status updated 51 seconds ago) DGMGRL> █</pre>
4.10	Let us check the properties of PRIMARY Database	• <code>show database MOIDDGDB_FL;</code> <pre>DGMGRL> show database NZDB_CHI; Database - nzdb_chi Role: PRIMARY Intended State: TRANSPORT-ON Instance(s): NZDB_CHI Database Status: SUCCESS</pre>
4.11	Let us check the properties of STANDBY Database	• <code>show database MOIDDGDB_NY;</code>
4.12	Check detailed database properties of PRIMARY DATABASE	<pre>DGMGRL> show database verbose MOIDDGDB_FL Database - MOIDDGDB_FL Role: PRIMARY</pre>

```

Intended State:      TRANSPORT-ON
Instance(s) :
    MOIDDGDB_FL

Properties:
    DGConnectIdentifier          = 'MOIDDGDB_FL'
    ObserverConnectIdentifier    = ''
    LogXptMode                   = 'ASYNC'
    RedoRoutes                   = ''
    DelayMins                    = '0'
    Binding                      = 'optional'
    MaxFailure                   = '0'
    MaxConnections               = '1'
    ReopenSecs                   = '300'
    NetTimeout                   = '30'
    RedoCompression              = 'DISABLE'
    LogShipping                   = 'ON'
    PreferredApplyInstance       = ''
    ApplyInstanceTimeout         = '0'
    ApplyLagThreshold            = '30'
    TransportLagThreshold        = '30'
    TransportDisconnectedThreshold = '30'
    ApplyParallel                = 'AUTO'
    ApplyInstances               = '0'
    StandbyFileManagement        = 'AUTO'
    ArchiveLagTarget             = '0'
    LogArchiveMaxProcesses       = '4'
    LogArchiveMinSucceedDest     = '1'
    DataGuardSyncLatency         = '0'
    DbFileNameConvert            = ''
    LogFileNameConvert           = ''
    FastStartFailoverTarget      = ''
    InconsistentProperties        = '(monitor)'
    InconsistentLogXptProps      = '(monitor)'
    SendQEntries                 = '(monitor)'
    LogXptStatus                 = '(monitor)'
    RecvQEntries                 = '(monitor)'
    PreferredObserverHosts       = ''
    StaticConnectIdentifier      = ''
' (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=Server-243) (PORT=1521)) (CONNECT_DATA=(SERVICE_NAME=
MOIDDGDB_FL_DGMGRL) (INSTANCE_NAME=MOIDDGDB_FL) (SERVER=DEDICATED))) '
    StandbyArchiveLocation       = '/u99/MOIDDGDB/Archive'
    AlternateLocation            = ''
    LogArchiveTrace              = '0'
    LogArchiveFormat             = 'arch_%t_%s_%r.arc'
    TopWaitEvents                = '(monitor)'

```

		Log file locations: Alert log : /u01/app/oracle/diag/rdbms/MOIDDGDB_FL/MOIDDGDB_FL/trace/alert_MOIDDGDB_FL.log Data Guard Broker log : /u01/app/oracle/diag/rdbms/MOIDDGDB_FL/MOIDDGDB_FL/trace/drcMOIDDGDB_FL.log Database Status: SUCCESS DGMGRL>																																						
4.13	Check detailed database properties of STANDBY DATABASE	DGMGRL> show database verbose MOIDDGDB_NY Database - MOIDDGDB_NY 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: 6.00 KByte/s Active Apply Rate: 351.00 KByte/s Maximum Apply Rate: 371.00 KByte/s Real Time Query: OFF Instance(s) : MOIDDGDB_NY Properties: <table><tr><td>DGConnectIdentifier</td><td>= 'MOIDDGDB_NY'</td></tr><tr><td>ObserverConnectIdentifier</td><td>= ''</td></tr><tr><td>LogXptMode</td><td>= 'ASYNC'</td></tr><tr><td>RedoRoutes</td><td>= ''</td></tr><tr><td>DelayMins</td><td>= '0'</td></tr><tr><td>Binding</td><td>= 'optional'</td></tr><tr><td>MaxFailure</td><td>= '0'</td></tr><tr><td>MaxConnections</td><td>= '1'</td></tr><tr><td>ReopenSecs</td><td>= '300'</td></tr><tr><td>NetTimeout</td><td>= '30'</td></tr><tr><td>RedoCompression</td><td>= 'DISABLE'</td></tr><tr><td>LogShipping</td><td>= 'ON'</td></tr><tr><td>PreferredApplyInstance</td><td>= ''</td></tr><tr><td>ApplyInstanceTimeout</td><td>= '0'</td></tr><tr><td>ApplyLagThreshold</td><td>= '30'</td></tr><tr><td>TransportLagThreshold</td><td>= '30'</td></tr><tr><td>TransportDisconnectedThreshold</td><td>= '30'</td></tr><tr><td>ApplyParallel</td><td>= 'AUTO'</td></tr><tr><td>ApplyInstances</td><td>= '0'</td></tr></table>	DGConnectIdentifier	= 'MOIDDGDB_NY'	ObserverConnectIdentifier	= ''	LogXptMode	= 'ASYNC'	RedoRoutes	= ''	DelayMins	= '0'	Binding	= 'optional'	MaxFailure	= '0'	MaxConnections	= '1'	ReopenSecs	= '300'	NetTimeout	= '30'	RedoCompression	= 'DISABLE'	LogShipping	= 'ON'	PreferredApplyInstance	= ''	ApplyInstanceTimeout	= '0'	ApplyLagThreshold	= '30'	TransportLagThreshold	= '30'	TransportDisconnectedThreshold	= '30'	ApplyParallel	= 'AUTO'	ApplyInstances	= '0'
DGConnectIdentifier	= 'MOIDDGDB_NY'																																							
ObserverConnectIdentifier	= ''																																							
LogXptMode	= 'ASYNC'																																							
RedoRoutes	= ''																																							
DelayMins	= '0'																																							
Binding	= 'optional'																																							
MaxFailure	= '0'																																							
MaxConnections	= '1'																																							
ReopenSecs	= '300'																																							
NetTimeout	= '30'																																							
RedoCompression	= 'DISABLE'																																							
LogShipping	= 'ON'																																							
PreferredApplyInstance	= ''																																							
ApplyInstanceTimeout	= '0'																																							
ApplyLagThreshold	= '30'																																							
TransportLagThreshold	= '30'																																							
TransportDisconnectedThreshold	= '30'																																							
ApplyParallel	= 'AUTO'																																							
ApplyInstances	= '0'																																							

		<pre> StandbyFileManagement = 'AUTO' ArchiveLagTarget = '0' LogArchiveMaxProcesses = '4' LogArchiveMinSucceedDest = '1' DataGuardSyncLatency = '0' DbFileNameConvert = '' LogFileNameConvert = '' FastStartFailoverTarget = '' InconsistentProperties = '(monitor)' InconsistentLogXptProps = '(monitor)' SendQEntries = '(monitor)' LogXptStatus = '(monitor)' RecvQEntries = '(monitor)' PreferredObserverHosts = '' StaticConnectIdentifier = '' '(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=server-233) (PORT=1521)) (CONNECT_DATA=(SERVICE_NAME= MOIDDGDB_NY_DGMGRL) (INSTANCE_NAME=MOIDDGDB_NY) (SERVER=DEDICATED)))' StandbyArchiveLocation = '/u99/MOIDDGDB/Archive' AlternateLocation = '' LogArchiveTrace = '0' LogArchiveFormat = 'arch_%t_%s_%r.arc' TopWaitEvents = '(monitor)' Log file locations: Alert log : /u01/app/oracle/diag/rdbms/MOIDDGDB_NY/MOIDDGDB_NY/trace/alert_MOIDDGDB_NY.log Data Guard Broker log : /u01/app/oracle/diag/rdbms/MOIDDGDB_NY/MOIDDGDB_NY/trace/drcMOIDDGDB_NY.log Database Status: SUCCESS DGMGRL> </pre>
4.14	Keep an eye on both database alert log files and lets perform a switchover.	• <code>switchover to MOIDDGDB_NY;</code>

		<pre> DGMGRL> switchover to NZDB_NY; Performing switchover NOW, please wait... Operation requires a connection to database "nzdb_ny" Connecting ... Connected to "NZDB_NY" Connected as SYSDBA. New primary database "nzdb_ny" is opening... Operation requires start up of instance "NZDB_CHI" on database "nzdb_chi" Starting instance "NZDB_CHI"... Connected to an idle instance. ORACLE instance started. Connected to "NZDB_CHI" Database mounted. Connected to "NZDB_CHI" Switchover succeeded, new primary is "nzdb_ny" DGMGRL> █ </pre>
4.15	Have you been watFLng alert log? Anything changed in there while database were switFLng its role? What is the output of the following SQL on both databases? SQL> @/tmp/dbrole.sql Did MOIDDGDB_FL became standby and did MOIDDGDB_NY became primary?	

```
SQL> @/tmp/dbrole.sql
```

```
NAME          OPEN_MODE    LOG_MODE    PROTECTION_MODE    DATABASE_ROLE
-----
NZDB          READ WRITE   ARCHIVELOG   MAXIMUM PERFORMANCE PRIMARY
```

```
SQL> !hostname
NY
```

```
SQL> @/tmp/dbrole.sql
```

```
NAME          OPEN_MODE    LOG_MODE    PROTECTION_MODE    DATABASE_ROLE
-----
NZDB          MOUNTED      ARCHIVELOG   MAXIMUM PERFORMANCE PHYSICAL STANDBY
```

```
SQL> !hostname
Chi
```

Part-5: Changing Data Guard Protection Modes

While keeping an eye on alert log, lets change Data Guard protection modes.

5.1	Connect to Data Broker	<code>dgmgrl sys/abc123@MOIDDGDB_FL</code>
5.2	For: Maximum Performance Mode	<pre>DGMGRL></pre> <code>edit configuration set protection mode as MaxPerformance;</code> <code>show configuration;</code>

```
DGMGRL> edit configuration set protection mode as MaxPerformance;
Succeeded.
DGMGRL> show configuration;

Configuration - nzdb_configuration

Protection Mode: MaxPerformance
Members:
  nzdb_ny - Primary database
  nzdb_chi - Physical standby database

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS (status updated 38 seconds ago)

DGMGRL> 
```

PRIMARY SQL> @/tmp/dbrole.sql

```
SQL> @/tmp/dbrole.sql

NAME          OPEN_MODE    LOG_MODE     PROTECTION_MODE    DATABASE_ROLE
-----
NZDB          READ WRITE   ARCHIVELOG    MAXIMUM PERFORMANCE PRIMARY

SQL> !hostname
NY
```

STANDBY SQL> @/tmp/dbrole.sql

```
SQL> @/tmp/dbrole.sql

NAME          OPEN_MODE    LOG_MODE     PROTECTION_MODE    DATABASE_ROLE
-----
NZDB          READ WRITE   ARCHIVELOG    MAXIMUM PERFORMANCE PRIMARY
```

5.3

For: Maximum Availability Mode

DGMGRL>

- show database MOIDDGDB_FL LogXptMode; → ASYNC
- show database MOIDDGDB_NY LogXptMode; → ASYNC
- edit database MOIDDGDB_FL set property LogXptMode=SYNC;
- edit database MOIDDGDB_NY set property LogXptMode=SYNC;
- edit configuration set protection mode as **MaxAvailability**;

```
DGMGRL> show database NZDB_CHI LogXptMode;
LogXptMode = 'ASYNC'
DGMGRL> show database NZDB_NY LogXptMode;
LogXptMode = 'ASYNC'
DGMGRL> edit database NZDB_CHI set property LogXptMode=SYNC;
Property "logxptmode" updated
DGMGRL> edit database NZDB_NY set property LogXptMode=SYNC;
Property "logxptmode" updated
DGMGRL> edit configuration set protection mode as MaxAvailability;
Succeeded.
```

- show configuration;

```
DGMGRL> show configuration;

Configuration - nzdb_configuration

Protection Mode: MaxAvailability
Members:
  nzdb_ny - Primary database
  nzdb_chi - Physical standby database

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS (status updated 28 seconds ago)
```

		<pre>PRIMARY SQL> @/tmp/dbrole.sql</pre> <pre>SQL> @/tmp/dbrole.sql</pre><table><thead><tr><th>NAME</th><th>OPEN_MODE</th><th>LOG_MODE</th><th>PROTECTION_MODE</th><th>DATABASE_ROLE</th></tr></thead><tbody><tr><td>NZDB</td><td>READ WRITE</td><td>ARCHIVELOG</td><td>MAXIMUM AVAILABILITY</td><td>PRIMARY</td></tr></tbody></table><pre>SQL> !hostname NY</pre> <pre>STANDBY SQL> @/tmp/dbrole.sql</pre> <pre>SQL> @/tmp/dbrole.sql</pre><table><thead><tr><th>NAME</th><th>OPEN_MODE</th><th>LOG_MODE</th><th>PROTECTION_MODE</th><th>DATABASE_ROLE</th></tr></thead><tbody><tr><td>NZDB</td><td>MOUNTED</td><td>ARCHIVELOG</td><td>MAXIMUM AVAILABILITY</td><td>PHYSICAL STANDBY</td></tr></tbody></table><pre>SQL> !hostname Chi</pre>	NAME	OPEN_MODE	LOG_MODE	PROTECTION_MODE	DATABASE_ROLE	NZDB	READ WRITE	ARCHIVELOG	MAXIMUM AVAILABILITY	PRIMARY	NAME	OPEN_MODE	LOG_MODE	PROTECTION_MODE	DATABASE_ROLE	NZDB	MOUNTED	ARCHIVELOG	MAXIMUM AVAILABILITY	PHYSICAL STANDBY
NAME	OPEN_MODE	LOG_MODE	PROTECTION_MODE	DATABASE_ROLE																		
NZDB	READ WRITE	ARCHIVELOG	MAXIMUM AVAILABILITY	PRIMARY																		
NAME	OPEN_MODE	LOG_MODE	PROTECTION_MODE	DATABASE_ROLE																		
NZDB	MOUNTED	ARCHIVELOG	MAXIMUM AVAILABILITY	PHYSICAL STANDBY																		
5.4	For: Maximum Protection Mode	<pre>DGMGRL></pre> • edit configuration set protection mode as MaxProtection; • show configuration;																				

```
DGMGRL> edit configuration set protection mode as MaxProtection;
Succeeded.
DGMGRL> show configuration;

Configuration - nzdb_configuration

Protection Mode: MaxProtection
Members:
  nzdb_ny - Primary database
  nzdb_chi - Physical standby database

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS (status updated 17 seconds ago)
```

PRIMARY SQL> @/tmp/dbrole.sql

```
SQL> @/tmp/dbrole.sql

NAME          OPEN_MODE    LOG_MODE    PROTECTION_MODE    DATABASE_ROLE
-----
NZDB          READ WRITE   ARCHIVELOG   MAXIMUM PROTECTION  PRIMARY

SQL> !hostname
NY
```

STANDBY SQL> @/tmp/dbrole.sql

```
SQL> @/tmp/dbrole.sql

NAME          OPEN_MODE    LOG_MODE    PROTECTION_MODE    DATABASE_ROLE
-----
NZDB          MOUNTED      ARCHIVELOG   MAXIMUM PROTECTION  PHYSICAL STANDBY

SQL> !hostname
Chi
```

Part-6: Data Broker Troubleshooting Commands

		<code>ALTER SYSTEM SET dg_broker_start=true; show parameter dg_broker_start</code>
		<code>dgmgrl sys/abc123@MOIDDGDB_FL</code>
		<code>CREATE CONFIGURATION MOIDDGDB_configuration AS PRIMARY DATABASE IS MOIDDGDB_FL CONNECT IDENTIFIER IS MOIDDGDB_FL;</code>
		<code>ADD DATABASE MOIDDGDB_NY AS CONNECT IDENTIFIER IS MOIDDGDB_NY MAINTAINED AS PHYSICAL;</code>
		<code>show configuration</code>
		<code>show database MOIDDGDB_FL</code>
		<code>show database verbose MOIDDGDB_FL</code>
		<code>edit configuration set protection mode as Maxperformance; edit configuration set protection mode as MaxAvailability; edit configuration set protection mode as MaxProtection;</code>
		<code>switchover to MOIDDGDB_FL</code>
		<code>show database 'MOIDDGDB_FL' InconsistentProperties</code>

Part-7: Errors and troubleshooting

No.	Problem	Resolution
7.1	DGMGRL> show configuration	DGMGRL> show database MOIDDGDB_FL

	Configuration - MOIDDGDB_configuration Protection Mode: MaxPerformance Members: MOIDDGDB_NY - Primary database MOIDDGDB_FL - Physical standby database Warning: ORA-16809: multiple warnings detected for the member	Database - MOIDDGDB_FL Role: PHYSICAL STANDBY Intended State: APPLY-ON Transport Lag: 27 minutes 14 seconds (computed 53 seconds ago) Apply Lag: (unknown) Average Apply Rate: (unknown) Real Time Query: OFF Instance(s): MOIDDGDB_FL Database Warning(s): ORA-16854: apply lag could not be determined ORA-16855: transport lag has exceeded specified threshold ORA-16857: member disconnected from redo source for longer than specified threshold Database Status: WARNING
7.2	ORA-16854: apply lag could not be determined	Check if standby is up Check if redo apply is running “ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;”

Hope this helps.

Moid Muhammad