exi

Oracle Database 19c

Data Guard Administration

Cd

# Oracle Net Services in a Data Guard Environment

## Overview

### Overview

In these practices, you will use graphical utilities to create and modify the Oracle network configuration files, and then propagate the resulting files to each server in the Data Guard environment.

## Practice 2-1: Configuring the tnsnames.ora File

### Overview

### Tasks

# Practices for Lesson 3: Creating a Physical Standby Database by Using SQL and RMAN Commands

## Practices for Lesson 3: Overview

### Practices Overview

## Practice 3-1: Prepare the Primary Database to Support Data Guard

### Overview

In this practice, you verify that the primary database is configured correctly to support a physical standby database. This should be done in the orclcdb env.

$ . oraenv

orclcdb

### Tasks

1. On localhost, invoke SQL\*Plus and connect as SYSDBA to your primary database . Determine if FORCE LOGGING is enabled. If it is not enabled, enable FORCE LOGGING mode.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Mon Jun 1 15:49:04 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **SELECT force\_logging FROM v$database;**

FORCE\_LOGGING NO

SQL> **ALTER DATABASE FORCE LOGGING;**

Database altered.

SQL> **SELECT force\_logging FROM v$database;**

FORCE\_LOGGING YES

1. Make sure the primary is in archivelog mode
   * Select log\_mode from v$database;
   * If the result is noarchivelog
   * Shutdown immediate;
   * Startup mount
   * Alter database archivelog;
   * Alter database open;
2. Determine the number of online redo log groups and their current size. Create standby redo log groups with one member for each group using the same size as the existing online redo logs. You should create one more additional group than the number you have for online redo log groups. Verify creation of the standby redo logs.

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

GROUP#

BYTES

1 209715200

2 209715200

3 209715200

SQL> alter database add standby logfile ('/u01/app/oracle/oradata/ORCLCDB/stdbyredo01.log') size 200M;

Database altered.

SQL> alter database add standby logfile ('/u01/app/oracle/oradata/ORCLCDB/stdbyredo02.log') size 200M;

Database altered.

SQL> alter database add standby logfile ('/u01/app/oracle/oradata/ORCLCDB/stdbyredo03.log') size 200M;

Database altered.

SQL> alter database add standby logfile ('/u01/app/oracle/oradata/ORCLCDB/stdbyredo04.log') size 200M;

Database altered.

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

GROUP# BYTES

1 209715200

2 209715200

3 209715200

SQL> **select group#,bytes from v$standby\_log;**

GROUP# BYTES

4 209715200

5 209715200

6 209715200

7 209715200

1. Define the first log archive destination to use the fast recovery area and enable it by using the set\_LAD\_1.sql script. Ensure that the changes are done both in memory and also stored the server parameter file. This location should be valid for any role and also valid for all types of log files.

SQL> alter system set log\_archive\_dest\_1='location=USE\_DB\_RECOVERY\_FILE\_DEST valid\_for=(ALL\_LOGFILES,ALL\_ROLES) db\_unique\_name=orclcdb' scope=both;

System altered.

SQL> alter system set log\_archive\_dest\_state\_1='enable' scope=both;

System altered.

1. Increase the maximum number of archive processes to 4.

SQL> **alter system set log\_archive\_max\_processes=4 scope=both;**

System altered.

**Note:** The documentation suggests this parameter be set to a value of 10. We are using a reduced number in this lab environment to reduce overhead.

1. Define the log\_archive\_config parameter to include entries for: orclcdb, stndby. Only stndby is needed at this moment, but the others can be added now in preparation for upcoming practices.
2. Enable automatic standby file management so that operating system file additions and deletions on the primary database are replicated to the standby database. This is normally set on the standby database. For the primary database, this is set for role reversals.

SQL> **alter system set standby\_file\_management='auto' scope=both;**

System altered.

## Practice 3-2: Prepare Host and Create Physical Standby Database

### Overview

In this practice, you will prepare stndby to receive the physical standby database and create the physical standby database using RMAN.

### Tasks

1. Use a terminal window logged in as oracle to stndby to create the initial directories needed for a physical standby database by running the crdir\_stndby script. Since the primary database is using multi-tenant architecture, additional directories are needed.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base has been set to /u01/app/oracle [oracle@stndby ~]$ **cat /home/oracle/setup/crdir\_stndby.sh**

...

mkdir -p /u01/app/oracle/admin/stndby/adump mkdir -p /u01/app/oracle/oradata/STNDBY

mkdir -p /u01/app/oracle/oradata/STNDBY/pdbseed mkdir -p /u01/app/oracle/oradata/STNDBY/dev1

mkdir -p /u01/app/oracle/fast\_recovery\_area/STNDBY [oracle@stndby ~]$ **/home/oracle/setup/crdir\_stndby.sh**

[oracle@stndby ~]$

1. Copy the initstandby.ora file to the ORACLE\_HOME/dbs directory.

[oracle@stndby ~]$ **cat /home/oracle/setup/create\_init.sh**

[oracle@stndby ~]$ **/home/oracle/setup/create\_init.sh**

[oracle@stndby ~]$

1. Verify the contents of the initstndby.ora file
2. Run /home/oracle/setup/create\_init.sh
3. Copy the password file from the primary host to the physical standby host by using the

scp\_pwdfile.sh script. Rename the file accordingly.

[oracle@stndby ~]$ **/home/oracle/setup/scp\_pwdfile.sh**

The authenticity of host 'localhost (192.0.2.11)' can't be established.

1. Set the environment to stndby
   * . oraenv
     + stndby
2. Start up nomount the stndby standby instance on stndby. This assumes that the terminal window you are using has previously set the environment variables to stndby. Exit SQL\*Plus when done.

[oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.3.0.1.0 Production on Tue Jun 11 03:51:01

2013

(c) 1982, 2019, Oracle. All rights reserved. Connected to an idle instance.

SQL> **startup nomount**

ORACLE instance started.

Total System Global Area 217157632 bytes Fixed Size 2286656 bytes Variable Size 159386560 bytes

Database Buffers 50331648 bytes

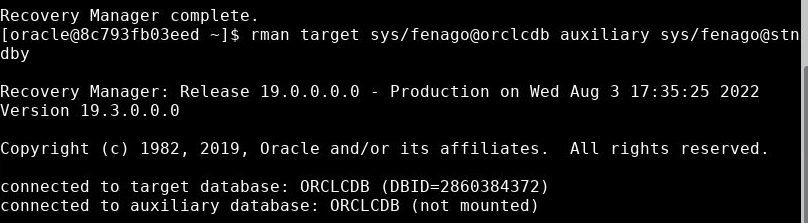
Redo Buffers 5152768 bytes SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production

Version 19.3.0.0.0

1. Create a physical standby on stndby by using the RMAN utility.
2. Connect using rman
   * rman target sys/fenago@orclcdb auxiliary sys/fenago@stndby



* + rman> duplicate target database for standby from active database;

## Practice 3-3: Start Redo Transport and Verify Operation

### Overview

.

### Tasks

1. Use a terminal window for localhost logged in as oracle with the environment variables set to orclcdb and start redo transport by defining log\_archive\_dest\_2 pointing to the physical standby database.

Alter system set log\_archive\_dest\_2=

Service="stndby", ASYNC NOAFFIRM db\_unique\_name="stndby", valid\_for=

(online\_logfile,all\_roles)

1. On the orclcdb database verify log shipping is enabled
   * sqlplus / as sysdba
   * @/home/oracle/setup/log\_ship
   * @/home/oracle/setup/gap\_status
     + This should say no gap. This is validating logs are being shipped and applied
2. On the stndby system, it may be best to open 2 terminal windows, one with the env or orclcdb and the other of stndby
   * . oraenv
     + Stndby
   * sqlplus / as sysdba
   * sql>@/home/oracle/setup/enable\_dg
     + This turns on managed recovery on the standby
   * sql>@/home/oracle/setup/mr.sql
     + This validates managed is running
3. On the primary or orclcdb issue
   * Alter system switch logfile;
   * Do this 3 times
   * Within sql then run @/home/oracle/setup/gap\_status

Graphical user interface

Description automatically generated with medium confidence

1. The Seq# may vary but no gap should be displayed

# Practices for Lesson 4: Managing Physical Standby Files After Structural Changes on the Primary Database

## Practices for Lesson 4: Overview

### Practices Overview

In these practices, you will test the primary database changes that do not require manual intervention at the standby database with the new features.

## Practice 4-1: Refreshing the Password File

### Overview

In this practice, you will test the automatic password change propagation feature. As of Oracle Database 19c Release 2 (19.3.0.1), password file changes done on the primary database are automatically propagated to standby databases.

### Tasks

1. Open a terminal window and connect to localhost as the oracle OS user.
2. Use the oraenv utility to set the environment variables for the orclcdb instance.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle

1. Invoke SQL\*Plus and connect as SYSDBA to your primary database.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Mon Jun 1 18:45:48 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 Production

Version 19.3.0.0.0

SQL>

1. List all users in the password file by using V$PWFILE\_USERS.

SQL> **col username format a10**

SQL> **SELECT username, sysdba, sysdg FROM v$pwfile\_users;**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USERNAME |  | SYSDB |  | SYSDG |
| SYS |  | TRUE |  | FALSE |

1. Open a new terminal window and use the SSH client to connect to stndby as the oracle

OS user.

1. Use the oraenv utility to set the environment variables for the stndby instance.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base has been set to /u01/app/oracle

1. Invoke SQL\*Plus and connect as SYSDBA to your primary database.

[oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Mon Jun 1 18:45:48 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL>

1. List all users in the password file by using V$PWFILE\_USERS.

SQL> **col username format a10**

SQL> **SELECT username, sysdba, sysdg FROM v$pwfile\_users;**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USERNAME |  | SYSDB |  | SYSDG |
| SYS |  | TRUE |  | FALSE |

1. Stop Media Recovery Process (mrp0) on purpose.

SQL> **alter database recover managed standby database cancel;**

Database altered.

1. Return to the terminal connected to primary database orclcdb and create a common user named C##DBA.

SQL> **CREATE USER c##dba IDENTIFIED BY <password> container=all;**

User created.

1. Grant SYSDBA and CREATE SESSION privileges to c##dba.

SQL> **GRANT sysdba, create session TO c##dba container=all;**

Grant succeeded.

1. Review the output of V$PWFILE\_USERS. The output shows that the newly created user was added to the password file.

SQL> **SELECT username, sysdba, sysdg FROM v$pwfile\_users;**

USERNAME SYSDB SYSDG

SYS C##DBA

TRUE FALSE TRUE FALSE

1. Return to the terminal session connected to stndby to review the output of V$PWFILE\_USERS. The new entry doesn’t appear in the output because the Media Recovery process was stopped.

SQL> **col username format a10**

SQL> **SELECT username, sysdba, sysdg FROM v$pwfile\_users;**

USERNAME SYSDB SYSDG

SYS

TRUE FALSE

1. Start the Media Recovery Process (mrp0).

SQL> **ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;**

Database altered.

1. Verify V$PWFILE\_USERS for the granted role.

SQL> **SELECT username, sysdba, sysdg FROM v$pwfile\_users;**

USERNAME SYSDB SYSDG

SYS TRUE FALSE C##DBA TRUE FALSE

**Note:** At times, the new entry doesn’t appear in the list immediately. In this case, run the

GRANT command (step 11) again at localhost to see if the new entry can be added.

1. Return to the terminal session connected to primary and change the password for C##DBA

user.

SQL> **ALTER USER c##dba IDENTIFIED BY Welcome\_1;**

User altered.

1. Test the connection to the standby database (stndby) with the new password.

SQL> **CONNECT c##dba/Welcome\_1@stndby as sysdba**

Connected.

1. Drop the c##dba user (primary).

SQL> **connect / as sysdba**

Connected

SQL> **DROP USER c##dba CASCADE;**

User dropped.

SQL>

1. Exit SQL\*Plus on localhost and stndby leaving the terminal windows open.

## Practice 4-2: Controlling PDB Replication

### Overview

In this practice, you will create two new PDBs (DEV2 and DEV3) in the primary database to demonstrate the control of the PDB replication to the standby database. First check to see if you have a dev1 pdb. If not use dbca to create dev1.

### Tasks

1. Use the terminal session on localhost. Create a directory for the new data files of DEV2.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [orclcdb] ? **orclcdb**

The Oracle base remains unchanged with value /u01/app/oracle [oracle@localhost ~]$ **mkdir -p /u01/app/oracle/oradata/ORCLCDB/dev2** [oracle@localhost ~]$

1. Invoke SQL\*Plus and connect to the CDB root as a user (SYS user in this practice) granted with CREATE PLUGGABLE DATABASE privilege to clone DEV2 from DEV1.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Mon Jun 1 21:58:28 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **CREATE PLUGGABLE DATABASE dev2 FROM dev1**

**CREATE\_FILE\_DEST='/u01/app/oracle/oradata/ORCLCDB/dev2';**

Pluggable database created. SQL>

1. Check the open mode of DEV2.

SQL> **show pdbs**

CON\_ID CON\_NAME

OPEN MODE RESTRICTED

1. PDB$SEED
2. DEV1
3. DEV2

READ ONLY NO READ WRITE NO MOUNTED

SQL>

1. Open DEV2 in READ WRITE mode.

SQL> **alter pluggable database DEV2 open;**

Pluggable database altered. SQL>

1. Use the terminal session connected to stndby. Connect as SYS to the stndby standby database.

[oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Mon Jun 1 22:08:54 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL>

1. List the value of the ENABLED\_PDBS\_ON\_STANDBY parameter.

**Note:** The asterisk (\*) indicates all PDBs are created and protected in the standby database.

SQL> **show parameter ENABLED\_PDBS\_ON\_STANDBY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME |  | TYPE |  | VALUE |
| enabled\_PDBs\_on\_standby SQL> |  | string |  | \* |

1. Verify the PDBs in the stndby standby database.

**Note:** The DEV2 PDB was successfully replicated to the standby database.

SQL> **show pdbs**

SQL>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CON\_ID |  | CON\_NAME |  | OPEN MODE |  | RESTRICTED |
| 2 |  | PDB$SEED |  | MOUNTED |  |  |
| 3 |  | DEV1 |  | MOUNTED |  |  |
| 4 |  | DEV2 |  | MOUNTED |  |  |

1. Now, let’s modify the ENABLED\_PDBS\_ON\_STANDBY parameter to include only DEV1 and

DEV2 PDBs in the standby database.

SQL> **alter system set ENABLED\_PDBS\_ON\_STANDBY = "DEV1","DEV2";**

System altered.

1. Return to the terminal session on localhost connected to the orclcdb database. Create a directory for DEV3.

SQL> **!mkdir -p /u01/app/oracle/oradata/ORCLCDB/dev3**

SQL>

1. Create DEV3 from DEV1.

SQL> **CREATE PLUGGABLE DATABASE dev3 FROM dev1**

**CREATE\_FILE\_DEST='/u01/app/oracle/oradata/ORCLCDB/dev3';**

Pluggable database created. SQL>

1. Check the open mode of DEV3.

SQL> **show pdbs**

CON\_ID CON\_NAME

OPEN MODE RESTRICTED

1. PDB$SEED
2. DEV1
3. DEV2
4. DEV3

READ ONLY NO READ WRITE NO READ WRITE NO MOUNTED

SQL>

1. Open DEV2 in READ WRITE mode.

SQL> **alter pluggable database DEV3 open;**

Pluggable database altered. SQL>

1. Return to the terminal session on stndby connected to the stndby database. Verify the PDBs in the standby database.

SQL> **show pdbs**

CON\_ID CON\_NAME

OPEN MODE RESTRICTED

1. PDB$SEED
2. DEV1
3. DEV2
4. DEV3

MOUNTED MOUNTED MOUNTED MOUNTED

1. The DEV3 PDB is listed in the output of the previous step, but it doesn’t meant that it is protected. Run the following query to check the recovery\_status column. This column shows whether recovery is enabled or disabled for the PDB.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SQL> **col name format a10**  SQL> **SELECT name, open\_mode, recovery\_status FROM v$pdbs;** | | | | |
| NAME |  | OPEN\_MODE |  | RECOVERY |
| PDB$SEED DEV1 DEV2 DEV3 |  | MOUNTED MOUNTED MOUNTED MOUNTED |  | ENABLED ENABLED ENABLED DISABLED |
| SQL> |  |  |  |  |

1. To clean up the environment, reset the ENABLED\_PDBS\_ON\_STANDBY parameter.

SQL> **ALTER SYSTEM SET enabled\_pdbs\_on\_standby="\*";**

System altered.

1. Return to the terminal session connected to localhost to clean up the environment.

SQL> **@/home/oracle/setup/cleanup04-02.sql**

Pluggable database altered. Pluggable database altered. Pluggable database dropped. Pluggable database dropped.

System altered.

SQL>

1. Exit SQL\*Plus on localhost and stndby leaving the terminal windows open for future practice.

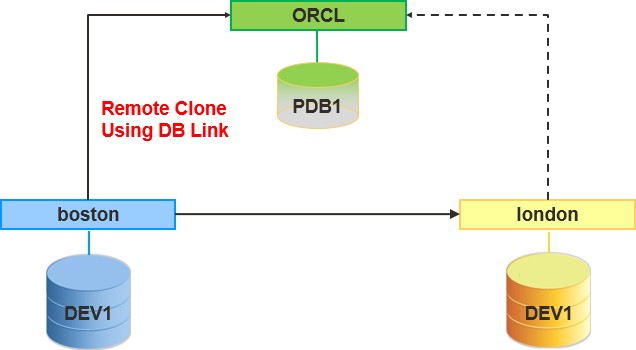
## Practice 4-3: Automating Instantiation of a PDB

### Overview

In this practice, you will test the usage of STANDBY\_PDB\_SOURCE\_FILE\_DBLINK to automate instantiation of a PDB in the stndby standby database when performing the remote PDB clone in the orclcdb primary database. In Oracle Database 19c, creating a PDB as a clone in the primary database requires copy of the data files belonging to the source PDB to the standby database manually.

### Assumptions

* + orclcdb: Primary database with a single PDB called DEV1
  + stndby: Standby database protecting the PDB called DEV1
  + ORCL: Stand-alone database with a single PDB called PDB1 as a remote clone source



### Prerequisites

* + The value of the STANDBY\_PDB\_SOURCE\_FILE\_DBLINK is only checked and used when a remote clone operation (create pluggable database….from pdb@dblink...) is executed on the primary database and the redo is applied at the standby database.
  + The standby database must be in Active Data Guard (ADG) mode. We require access to the dictionary for the database link and the dictionary is only available if the standby is in Active Data Guard mode.
  + The source PDB must be in Read Only mode and remain for the duration of the copies to the primary and all standby databases in the configuration. We do not at this time support automatic maintenance of the standby database of any type with the 12.2 hot cloning or relocate features.

### Tasks

1. Open a terminal window and use the SSH client to connect to em13c as the oracle OS user. Check the status of the precreated ORCL database and its PDB1 PDB. If it’s not running, start up the database.

Last login: Mon Jun 8 09:30:39 2020 [oracle8c793fb03eed ~]$ **. oraenv** ORACLE\_SID = [ORCL] ? **ORCL**

The Oracle base remains unchanged with value /u01/app/oracle [oracle8c793fb03eed ~]$

[oracle8c793fb03eed ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Fri Jun 19 16:54:24 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **show pdbs**

CON\_ID CON\_NAME

OPEN MODE RESTRICTED

1. PDB$SEED
2. PDB1 SQL>

READ ONLY NO

READ WRITE NO

1. Create the database link user in the ORCL database.

SQL> **CREATE USER c##remote\_user IDENTIFIED BY <password> CONTAINER=all;**

User created.

SQL> **GRANT CREATE SESSION, CREATE PLUGGABLE DATABASE TO**

**c##remote\_user CONTAINER=ALL;**

Grant succeeded.

SQL>

1. Use the terminal window connected to localhost and create a database link in the orclcdb

primary database to the ORCL source database using the database link.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Tue Jun 2 10:12:34 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **CREATE DATABASE LINK clone\_link CONNECT TO c##remote\_user IDENTIFIED BY <password> USING 'ORCL'**;

Database link created. SQL>

1. Test the database link from the orclcdb primary database to the ORCL source database.

SQL> **SELECT \* FROM dual@clone\_link;**

D

- X

SQL>

1. Use the terminal window connected to stndby and configure the stndby database in Active Data Guard with the real-time query mode by executing the configure\_adg.sql script.

**Note:** The real-time query feature is covered in the next lesson.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base has been set to /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Tue Jun 2 10:22:21 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **@/home/oracle/setup/configure\_adg.sql**

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;

Database altered.

SQL> ALTER DATABASE OPEN READ ONLY;

Database altered.

sql

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION;

Database altered.

1. Test the database link from the stndby standby database to the ORCL source database and set the STANDBY\_PDB\_SOURCE\_FILE\_DBLINK parameter.

SQL> **SELECT \* FROM dual@clone\_link;**

D

- X

SQL> **ALTER SYSTEM SETSTANDBY\_PDB\_SOURCE\_FILE\_DBLINK='clone\_link';**

System altered.

1. Return to the ORCL terminal session on localhostalt and open PDB1 in read-only mode.

SQL> **ALTER PLUGGABLE DATABASE pdb1 CLOSE;**

Pluggable database altered.

SQL> **ALTER PLUGGABLE DATABASE pdb1 OPEN READ ONLY;**

Pluggable database altered. SQL>

1. Return to the orclcdb terminal session on localhost and create a new pluggable database called NEW\_PDB1 as a clone of the remote pluggable database PDB1.

SQL> **ALTER SESSION SET**

**db\_create\_file\_dest='/u01/app/oracle/oradata/ORCLCDB';**

Session altered.

SQL> **CREATE PLUGGABLE DATABASE new\_pdb1 FROM pdb1@clone\_link;**

Pluggable database created. SQL>

1. Open a new terminal window connected to stndby and review the alert log file.

oracle@stndby's password: *<password>*

Last login: Sun Jun 14 17:33:09 2020 from gateway.example.com [oracle@stndby ~]$ **tail -100**

**/u01/app/oracle/diag/rdbms/stndby/stndby/trace/alert\_stndby.log**

...

2020-06-02T17:05:09.757443-04:00

Recovery created pluggable database NEW\_PDB1

...

Recovery attempting to copy datafiles for pdb-NEW\_PDB1 from source pdb-PDB1 at dblink-clone\_link

...

1. Return to the ORCL terminal session on 8c793fb03eed. You can now safely open PDB1 in read write mode.

SQL> **ALTER PLUGGABLE DATABASE pdb1 CLOSE;**

Pluggable database altered.

SQL> **ALTER PLUGGABLE DATABASE pdb1 OPEN;**

Pluggable database altered.

1. Return to the orclcdb terminal session on localhost and open the NEW\_PDB1 PDB. **Note:** You may observe a warning message. It’s probably because of the mismatched settings in the source CDB (ORCL) and primary database . You can safely ignore it.

SQL> **ALTER PLUGGABLE DATABASE NEW\_PDB1 open;**

Pluggable database altered.

SQL> **col name format a10**

SQL> **SELECT name, open\_mode, recovery\_status FROM v$pdbs;**

NAME

OPEN\_MODE RECOVERY

PDB$SEED DEV1

NEW\_PDB1

READ ONLY

ENABLED

READ WRITE ENABLED

READ WRITE ENABLED

SQL>

1. Return to terminal session connected to the stndby database on stndby and reset the parameter and check the status of NEW\_PDB1 PDB.

SQL> **ALTER SYSTEM RESET standby\_pdb\_source\_file\_dblink;**

System altered.

SQL> **col name format a10**

SQL> **SELECT name, open\_mode, recovery\_status FROM v$pdbs;**

|  |  |  |
| --- | --- | --- |
| NAME | OPEN\_MODE | RECOVERY |
| PDB$SEED | READ ONLY | ENABLED |
| DEV1 | MOUNTED | ENABLED |
| NEW\_PDB1 | MOUNTED | ENABLED |
| SQL> |  |  |

1. Return to the orclcdb terminal session on localhost to clean up the environment.

SQL> **ALTER PLUGGABLE DATABASE new\_pdb1 CLOSE;**

Pluggable database altered.

SQL> **DROP PLUGGABLE DATABASE new\_pdb1 INCLUDING DATAFILES;**

Pluggable database dropped. SQL>

1. Exit SQL\*Plus on localhost, and stndby leaving the terminal windows open for future practices.

Congratulations! In this practice, you tested how to automate the instantiation of a PDB in the standby database using the STANDBY\_PDB\_SOURCE\_FILE\_DBLINK parameter when performing a remote clone in the primary database.

# Practices for Lesson 5: Using Oracle Active Data Guard: Supported Workloads in Read-Only Standby

## Practices for Lesson 5: Overview

### Practices Overview

In these practices, you will configure the Active Data Guard standby databases to support the alte@various offloadable workloads such as real-time query, DML/DDL on Global Temporary Tables, and read-mostly applications.

1. Set parameter temp\_undo\_enable=true;

**alter system set adg\_redirect\_dml=true scope=both;**

System altered.

SQL> **show parameter adg\_redirect\_dml**

## Practice 5-1: Enable Active Data Guard Real-Time Query

### Overview

In this practice, you enable the Active Data Guard with the real-time query feature and verify its operation.

### Tasks

1. Use a terminal window logged in as oracle to stndby with the environment variables set for stndby appropriately. Make sure that the physical standby database and its DEV1 PDB are in READ ONLY mode.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base has been set to /u01/app/oracle [oracle@stndby ~]$

[oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Tue Jun 2 20:50:44 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **show pdbs**

CON\_ID CON\_NAME

OPEN MODE RESTRICTED

1. PDB$SEED
2. DEV1

READ ONLY NO

MOUNTED

SQL> **alter pluggable database DEV1 open;**

Pluggable database altered.

SQL>

1. (**Optional**) If the physical standby database is in the MOUNT state, stop the redo apply service and open the stndby database in READ ONLY mode.

SQL> **alter database open read only;**

alter database open read only

\*

ERROR at line 1:

ORA-10456: cannot open standby database; media recovery session may be in

progress

SQL>

SQL> **alter database recover managed standby database cancel;**

Database altered.

SQL> **alter database open read only;**

Database altered.

SQL> **alter pluggable database DEV1 open;**

Pluggable database altered. SQL>

1. Restart the Redo Apply process on the physical standby database running in the READ ONLY mode to enable the real-time query feature.

**Note:** Depending on the Redo Apply process status, you will see one of two results.

SQL> **alter database recover managed standby database disconnect;**

Database altered.

**OR**

SQL> **alter database recover managed standby database disconnect;**

alter database recover managed standby database disconnect

\*

ERROR at line 1:

ORA-01153: an incompatible media recovery is active

SQL>

1. This database is using the Oracle Multitenant option. The default operating system authentication method for the multitenant container database (CDB) is to the container root (CDB$ROOT). Data Guard environment operations are performed at the CDB level. Schema objects like the sample schemas exist in customer created pluggable databases (PDBs). Verify that the SQL\*Plus session is currently connected to the CDB$ROOT and that sample schemas do not exist in the root container. Two ways are illustrated to determine the current container name. The first technique uses the SQL\*Plus show commands. The second technique uses all SQL syntax. The HR.REGIONS table is part of the sample schemas, but should not exist in the root container.

SQL> **show con\_id**

CON\_ID 1

SQL> **show con\_name**

CON\_NAME CDB$ROOT

SQL> **SELECT sys\_context ('USERENV', 'CON\_NAME') FROM dual;**

SYS\_CONTEXT('USERENV','CON\_NAME') CDB$ROOT

SQL> **select \* from hr.regions;**

select \* from hr.regions

\*

ERROR at line 1:

ORA-00942: table or view does not exist

1. Switch the SQL\*Plus session to the DEV1 pluggable database (PDB) and query the

HR.REGIONS table again.

SQL> **ALTER SESSION SET CONTAINER = DEV1;**

Session altered.

SQL> **select \* from hr.regions;**

REGION\_ID REGION\_NAME

1. Europe
2. Americas
3. Asia
4. Middle East and Africa

SQL>

1. Leave the above window open. Open a terminal window (if not already open) logged in as oracle to localhost with the environment variables set for orclcdb appropriately. Launch SQL\*Plus and switch session to the DEV1 PDB of the primary database. Query the HR.REGIONS table.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Tue Jun 2 20:50:44 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter session set container = DEV1;**

Session altered.

SQL> **select \* from hr.regions order by region\_id;**

REGION\_ID REGION\_NAME

1. Europe
2. Americas
3. Asia
4. Middle East and Africa
5. Insert a new row into the HR.REGIONS table and commit the SQL statement.

SQL> **insert into hr.regions values (5,'Australia');**

1 row created.

SQL> **commit;**

Commit complete.

1. Return to the SQL\*Plus session to the DEV1 PDB that is still open for the physical standby database on stndby and query the HR.REGIONS value. The new row is immediately available on the physical standby database for reporting applications after it was inserted on the primary database. This illustrates the real-time query capability of Active Data Guard.

SQL> **select \* from hr.regions order by region\_id;**

REGION\_ID REGION\_NAME

1. Europe
2. Americas
3. Asia
4. Middle East and Africa
5. Australia
6. Exit SQL\*Plus on stndby of the physical standby database. It is recommended to keep the terminal session open with the environment variables set appropriately.
7. Exit SQL\*Plus on localhost, leaving the window open for future practices.

## Practice 5-2: Performing DDL/DML on Global Temporary Table

### Overview

In this practice, you will issue DML and DDL operations on a global temporary table in the

stndby standby database and verify its operations.

This feature benefits Oracle Data Guard in the following ways:

* + Read-mostly reporting applications that use global temporary tables for storing temporary data can be offloaded to an Oracle Active Data Guard instance.
  + When temporary undo is enabled on the primary database, undo for changes to a global temporary table are not logged in the redo and thus, the primary database generates less redo. Therefore, the amount of redo that Oracle Data Guard must ship to the standby is also reduced, thereby reducing network bandwidth consumption and storage consumption.

### Tasks

1. Use a terminal window logged in as oracle to stndby with the environment variables set for stndby appropriately. Check if the real time query is enabled in the stndby database as the system user.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base has been set to /u01/app/oracle [oracle@stndby ~]$ **sqlplus system/*<password>***

SQL\*Plus: Release 19.0.0.0.0 - Production on Wed Jun 3 00:10:07 2020

Version 19.3.0.0.0

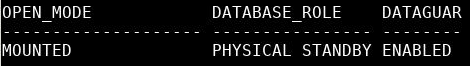
(c) 1982, 2019, Oracle. All rights reserved. Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **select OPEN\_MODE, DATABASE\_ROLE, DATAGUARD\_BROKER from**

**v$database;**



1. Attempt to create a Global Temporary Table (GTT) on Active Data Guard (read-only) instance. NOTE: In 19c this is done on the primary database and it moves it to the stndby. So create global table on primary and you will see it on stndby.

SQL> **CREATE GLOBAL TEMPORARY TABLE gtt01 (c1 number, c2 varchar2(10)) ON COMMIT PRESERVE ROWS;**

Table created.

SQL>

**Note:** The DDL operation on the global temporary table is redirected to the primary database. DDL change is visible on the standby database when it catches up with the primary database.

1. Use a terminal window logged in as oracle to localhost with the environment variables set for orclcdb appropriately. Log in as system to the orclcdb database.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? orclcdb

The Oracle base has been set to /u01/app/oracle [oracle@localhost ~]$ **sqlplus system/*<password>***

SQL\*Plus: Release 19.0.0.0.0 - Production on Tue Jun 2 22:48:02 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

1. Check if the global temporary table called GTT01 was created in the primary database.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SQL> **DESC** | **gtt01** |  | | | |
| Name |  |  | Null? |  | Type |
| C1 C2 |  |  |  |  | NUMBER VARCHAR2(10) |

**Note:** As you can see, the DDL operation was redirected and issued in the primary database.

1. Return to the stndby terminal session on stndby and check the TEMP\_UNDO\_ENABLED

parameter.

|  |  |  |  |
| --- | --- | --- | --- |
| SQL> **show parameter TEMP\_UNDO\_ENABLED** | | | |
| NAME | TYPE |  | VALUE |
| temp\_undo\_enabled SQL> | boolean |  | FALSE |

1. Now, attempt to insert a row in the global temporary table on the standby database.

SQL> **INSERT INTO gtt01 VALUES(10,'ABC');**

1 row created.

SQL> **COMMIT;**

Commit complete.

SQL>

**Note:** Even if the TEMP\_UNDO\_ENABLED was set to FALSE, the INSERT statement in the physical standby database was allowed. The TEMP\_UNDO\_ENABLED parameter is only applicable for the primary database. For a standby database, this parameter is ignored because temporary undo is enabled by default on the standby database.

1. Exit SQL\*Plus on localhost and stndby, leaving the window open for future practices.

## Practice 5-3: Managing Private Temporary Table for DDL/DML

### Overview

In this practice, you will create the private temporary tables in the stndby physical standby database and test the DML/DDL operations in the private temporary tables.

Private temporary tables are useful in the following situations:

* + When an application stores temporary data in transient tables that are populated once, read few times, and then dropped at the end of a transaction or session
  + When a session is maintained indefinitely and must create different temporary tables for different transactions
  + When the creation of a temporary table must not start a new transaction or commit an existing transaction
  + When different sessions of the same user must use the same name for a temporary table
  + When a temporary table is required for a read-only database

### Tasks

1. Use a terminal window logged in as oracle to stndby with the environment variables set for stndby appropriately. Log in as the SYSTEM user.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base has been set to /u01/app/oracle [oracle@stndby ~]$ **sqlplus system/*<password>***

SQL\*Plus: Release 19.0.0.0.0 - Production on Wed Jun 3 11:24:59 2020

Version 19.3.0.0.0

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

Last Successful login time: Wed Jun 03 2020 10:44:46 -04:00 Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0 SQL>

1. Attempt to create a private temporary table (PTT).

SQL> **CREATE PRIVATE TEMPORARY TABLE mine (x NUMBER, y VARCHAR2(10));**

CREATE PRIVATE TEMPORARY TABLE mine (x NUMBER, y VARCHAR2(10))

\*

ERROR at line 1:

ORA-00903: invalid table name

1. Check the value of the PRIVATE\_TEMP\_TABLE\_PREFIX parameter.

SQL> **show parameter PRIVATE\_TEMP\_TABLE\_PREFIX**

NAME

TYPE

VALUE

private\_temp\_table\_prefix string

ORA$PTT\_

1. Create a PTT with the appropriate prefix.

SQL> **CREATE PRIVATE TEMPORARY TABLE ora$ptt\_mine (x NUMBER, y VARCHAR2(10));**

Table created.

**Note:** Because the definition of a private temporary table is stored in memory, you can

create it in the read only standby database. But the table name must include the appropriate prefix.

1. Insert rows in the PTT.

SQL> **INSERT INTO ora$ptt\_mine VALUES (1,'Work1');**

1 row created.

1. Display data from the PTT.

SQL> **SELECT \* FROM ora$ptt\_mine;**

X Y

1 Work1

1. Find all information related to the PTT using the show\_ptt.sql script.

SQL> **@/home/oracle/setup/show\_ptt.sql**

...

SQL> SELECT sid, serial#, table\_name, tablespace\_name, duration FROM dba\_private\_temp\_tables;

SID

SERIAL# TABLE\_NAME

TABLESPACE\_NAME DURATION

237

58512 ORA$PTT\_MINE

TEMP

TRANSACTION

1. Open another terminal window logged in as oracle to stndby with the environment variables set for stndby appropriately. Log in as the SYSTEM user.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base has been set to /u01/app/oracle [oracle@stndby ~]$ **sqlplus system/*<password>***

SQL\*Plus: Release 19.0.0.0.0 - Production on Wed Jun 3 11:24:59 2020

Version 19.3.0.0.0

(c) 1982, 2019, Oracle. All rights reserved. Last Successful login time: Wed Jun 03 2020 10:44:46 -04:00 Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0 SQL>

1. Verify that the PTT created by the first SYSTEM session is not visible to the second system

session.

SQL> **desc ORA$PTT\_MINE**

ERROR:

ORA-04043: object ORA$PTT does not exist

1. Return to the first SYSTEM window session on stndby and issue the ROLLBACK statement.

SQL> **ROLLBACK;**

Rollback complete.

SQL> **@/home/oracle/setup/show\_ptt.sql**

...

SQL> SELECT sid, serial#, table\_name, tablespace\_name, duration FROM dba\_private\_temp\_tables;

no rows selected SQL>

**Note:** The duration of the ORA$PTT\_MINE table was TRANSACTION. This is the default duration type. This means that the PTT is automatically dropped at the end of the transaction in which the PTT has been created.

1. In the same terminal window, create a new PTT of SESSION duration type that will last until your session ends.

SQL> **CREATE PRIVATE TEMPORARY TABLE ora$ptt\_mine2 (x NUMBER, y VARCHAR2(10)) ON COMMIT PRESERVE DEFINITION;**

Table created.

1. Find all information related to the PTT.

SQL> **@/home/oracle/setup/show\_ptt.sql**

...

SQL> SELECT sid, serial#, table\_name, tablespace\_name, duration FROM dba\_private\_temp\_tables;

SID

SERIAL# TABLE\_NAME

TABLESPACE\_NAME DURATION

237

58512 ORA$PTT\_MINE2 TEMP

SESSION

1. Insert rows in the PTT.

SQL> **INSERT INTO ora$ptt\_mine2 VALUES (2,'Work2');**

1 row created.

1. Display data from the PTT.alaltersalte

SQL> **SELECT \* FROM ora$ptt\_mine2;**

X Y

2 Work2

1. Now, issue the COMMIT statement and display the information about PTT.

SQL> **COMMIT;**

Commit complete.

SQL> **@/home/oracle/setup/show\_ptt.sql**

...

SQL> SELECT sid, serial#, table\_name, tablespace\_name, duration FROM dba\_private\_temp\_tables;

SID

SERIAL# TABLE\_NAME

TABLESPACE\_NAME DURATION

237

58512 ORA$PTT\_MINE2 TEMP

SESSION

**Note:** The PTT still exists. It will be dropped at the end of the session.

1. Exit SQL\*Plus on localhost and stndby, leaving the window open for future practices.

## Practice 5-4: Configuring Automatic Redirection of DML operations

### Overview

In this practice, you will enable automatic redirection of DML operations for standby sessions in an Active Data Guard environment to support read-mostly applications, which occasionally execute DMLs, on the standby database.

### Tasks

1. Use a terminal window logged in as oracle to localhost with the environment variables set for orclcdb appropriately. Log in as the SYS user.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [orclcdb] ? **orclcdb**

The Oracle base remains unchanged with value /u01/app/oracle [oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Wed Jun 3 13:20:37 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL>

1. Configure automatic redirection of DML operations in the primary database.

SQL> **alter system set adg\_redirect\_dml=true scope=both;**

System altered.

SQL> **show parameter adg\_redirect\_dml**

NAME

TYPE

VALUE

adg\_redirect\_dml SQL>

boolean

TRUE

1. Use a terminal window logged in as oracle to stndby with the environment variables set for stndby appropriately. Log in as the SYS user.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [stndby] ? **stndby**

The Oracle base remains unchanged with value /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Wed Jun 3 13:24:56 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL>

1. Configure automatic redirection of DML operations in the standby database.

SQL> **alter system set adg\_redirect\_dml=true scope=both;**

System altered.

SQL> **show parameter adg\_redirect\_dml**

NAME

TYPE

VALUE

adg\_redirect\_dml SQL>

boolean

TRUE

1. Return to the terminal session connected to localhost. Connect to the DEV1 PDB for testing.

SQL> **alter session set container=DEV1;**

Session altered. SQL> **show con\_name**

CON\_NAME

DEV1 SQL>

1. Create a table called TEST01 and insert a row.

SQL> **CREATE TABLE test01 (id number(10), name varchar2(10));**

Table created.

SQL> **INSERT INTO test01 VALUES(10, 'SEAN');**

1 row created.

SQL> **COMMIT;**

Commit complete.

SQL> **SELECT \* FROM test01;**

ID NAME

10 SEAN

1. Return to the terminal session connected to stndby. Connect to the DEV1 PDB.

SQL> **alter session set container = DEV1;**

Session altered. SQL> **show con\_name** CON\_NAME

DEV1

1. Display the data in the TEST01 table.

SQL> **SELECT \* FROM test01;**

ID NAME

10 SEAN

1. Test automatic redirection of DML in the current session.

SQL> **DELETE FROM test01;**

DELETE FROM test01

\* ERROR at line 1:

ORA-16397: statement redirection from Oracle Active Data Guard standby database

to primary database failed

SQL> **!oerr ora 16397**

16397, 00000, "statement redirection from Oracle Active Data Guard standby database to primary database failed"

// \*Cause: The statement redirection failed because of one of the following reasons:

//

//

//

//

// temporary

//

//

1. The primary database connect string was not established.
2. The primary database could not be reached.
3. The undo-mode or incarnation were not the same.
4. The current user and logged-in user were not the same.
5. Redirecting CREATE TABLE AS SELECT (CTAS) of the global

table was not supported.

6. Redirecting PL/SQL execution having bind variable was not

supported.

// \*Action: Run the statement after fixing the condition that caused the failure.

**Note:** You need to log in to the DEV1 PDB using username/password instead of the ALTER

SESSION SET CONTAINER command.

1. Exit SQL\*Plus and log in to the DEV1 PDB again and test automatic redirection of DML.

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0 [oracle@stndby ~]$ **sqlplus**

**sys/<password>@stndby:1521/dev1.example.com as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Wed Jun 3 13:42:48 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **DELETE FROM test01;**

1 row deleted.

SQL> **COMMIT;**

Commit complete.

1. Return to the terminal session connected to localhost. Verify the result of the automatic redirection of DML feature in the orclcdb primary database.

SQL> **SELECT \* FROM test01;**

no rows selected

**Note:** The DELETE statement issued from the standby database was redirected to the

primary database.

1. Exit SQL\*Plus on localhost and stndby leaving the terminal windows open for future practices.

# Practices for Lesson 6: Using Oracle Active Data Guard: Far Sync and Real-Time Cascading

## Practices for Lesson 6: Overview

### Practices Overview

In these practices, you will implement two Far Sync instances into the current Data Guard environment.

## Practice 6-1: Add Far Sync to the Data Guard Environment

### Overview

In this practice, you will create a Far Sync instance (orclcdbFS) on host02 that is in close proximity to the primary database. THIS CLASS doesn't have another network to connect too so this information is for those who have the multiple networks to utilize

### Tasks – This is a read only chapter as we do not have another host.

1. Use a terminal window for localhost logged in as oracle with the environment variables set to orclcdb. Use SQL\*Plus to create a text-based initialization parameter file named /tmp/initorclcdbFS.ora that contains a copy of all the current parameters for primary primary

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Wed Jun 3 15:45:55 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **create pfile='/tmp/initorclcdbFS.ora' from spfile;**

File created.

1. Create a Far Sync control file named /tmp/orclcdbFS.ctl on the primary database and exit SQL\*Plus when done.

SQL> **alter database create far sync instance controlfile as '/tmp/orclcdbFS.ctl';**

Database altered. SQL> **exit**

1. Copy the primary password file to the /tmp directory.

[oracle@localhost ~]$ **cp $ORACLE\_HOME/dbs/orapworclcdb /tmp**

[oracle@localhost ~]$

1. Transfer the three files staged in the /tmp directory from localhost to host02 and place them into the /tmp directory on host02.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [oracle@localhost ~]$ **cat /home/oracle/setup/copy\_orclcdbFS.sh**  ...  scp /tmp/initorclcdbFS.ora oracle@host02:/tmp scp /tmp/orclcdbFS.ctl oracle@host02:/tmp  scp /tmp/orapworclcdb oracle@host02:/tmp  [oracle@localhost ~]$ **/home/oracle/setup/copy\_orclcdbFS.sh** | | | | |
| oracle@host02's password: | *<password>* |  |  |  |
| initorclcdbFS.ora | 100% | 1718 | 1.6MB/s | 00:00 |
| oracle@host02's password: | *<password>* |  |  |  |
| orclcdbFS.ctl | 100% | 18MB | 33.5MB/s | 00:00 |
| oracle@host02's password: | *<password>* |  |  |  |
| orapworclcdb | 100% | 2560 | 2.6MB/s | 00:00 |
| [oracle@localhost ~]$ |  |  |  |  |

1. Open a terminal window for host02 logged in as oracle with the environment variables set to orclcdbFS. Create the initial directories needed on the Far Sync server. These are the same directories that were created on the physical standby server stndby in practice 3-2.

[oracle@host02 ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdbFS**

The Oracle base has been set to /u01/app/oracle [oracle@host02 ~]$ **cat /home/oracle/setup/crdir\_host02.sh** mkdir -p /u01/app/oracle/admin/orclcdbFS/adump

mkdir -p /u01/app/oracle/oradata/orclcdbFS

mkdir -p /u01/app/oracle/oradata/orclcdbFS/pdbseed mkdir -p /u01/app/oracle/oradata/orclcdbFS/dev1

mkdir -p /u01/app/oracle/fast\_recovery\_area/orclcdbFS

**Note:** Linux directory and file names are case sensitive. Throughout these labs, the names for Far Sync use the format orclcdbFS and stndbyFS for readability. The last 2 letters are in upper-case.

* 1. The entry for log\_archive\_dest\_2 uses the valid\_for option assuming it has the role of primary database. On the Far Sync, this needs to use the role of a standby database. Also the Far Sync should use ASYNC communication to the physical standby environment. Correct the log\_archive\_dest\_2 parameter to the following value (Changes to make in bold):

\*.log\_archive\_dest\_2='SERVICE=stndby **ASYNC** REOPEN=15 valid\_for=(**STANDBY\_LOGFILES**,**STANDBY\_ROLE**) db\_unique\_name=stndby'

* 1. The original control files are named control01.ctl and control02.ctl. We will rename these orclcdbFS01.ctl and orclcdbFS02.ctl, respectively. This is for preference only since these files are not normal control files. Both changes can be made with the following global search and replace:

**:%s/control0/orclcdbFS0/g**

* 1. Add the following new entries to the bottom of the file.

**\*.db\_unique\_name=orclcdbFS**

**\*.fal\_server=orclcdb**

**\*.log\_file\_name\_convert='ORCLCDB','orclcdbFS'**

* 1. Remove or comment out the LOCAL\_LISTENER entry if it exists with the value.

**#**\*.local\_listener='LISTENER\_orclcdbFS'

* 1. Recheck all modifications and case-sensitivity issues. If accurate, then save the changes made to the file.

**:wq!**

* 1. The complete edited file is listed below for reference. Bold entries indicate changes that were made.

[oracle@host02 ~]$ **cat /tmp/initorclcdbFS.ora** orclcdbFS. data\_transfer\_cache\_size=0 orclcdbFS. db\_cache\_size=46137344

orclcdbFS. java\_pool\_size=12582912 orclcdbFS. large\_pool\_size=12582912

orclcdbFS. oracle\_base='/u01/app/oracle'#ORACLE\_BASE set from environment

orclcdbFS. pga\_aggregate\_target=209715200 orclcdbFS. sga\_target=310378496

orclcdbFS. shared\_io\_pool\_size=4194304 orclcdbFS. shared\_pool\_size=226492416 orclcdbFS. streams\_pool\_size=0

\*.audit\_file\_dest='/u01/app/oracle/admin/**orclcdbFS**/adump'

\*.audit\_trail='db'

\*.compatible='12.1.0.0.0'

\*.control\_files='/u01/app/oracle/oradata/**orclcdbFS**/**orclcdbFS01**.ctl ','/u01/app/oracle/fast\_recovery\_area/**orclcdbFS**/**orclcdbFS02**.ctl'

\*.db\_block\_size=8192

\*.db\_domain='example.com'

\*.db\_name='**orclcdb**'

\*.db\_recovery\_file\_dest='/u01/app/oracle/fast\_recovery\_area'

\*.db\_recovery\_file\_dest\_size=10g

\*.diagnostic\_dest='/u01/app/oracle'

\*.log\_archive\_config='dg\_config=(**orclcdb**,**orclcdbFS**,stndby,stndbyFS

,stndby2)'

\*.log\_archive\_dest\_1='location=USE\_DB\_RECOVERY\_FILE\_DEST valid\_for=(ALL\_LOGFILES,ALL\_ROLES) db\_unique\_name=**orclcdbFS**'

\*.log\_archive\_dest\_2='SERVICE=stndby **ASYNC** REOPEN=15 valid\_for=(**STANDBY\_LOGFILES,STANDBY\_ROLE**) db\_unique\_name=stndby'

\*.log\_archive\_dest\_state\_1='enable'

\*.log\_archive\_format='arch\_%t\_%s\_%r.log'

\*.log\_archive\_max\_processes=4

\*.memory\_target=496m

\*.open\_cursors=300

\*.processes=300

\*.remote\_login\_passwordfile='EXCLUSIVE'

\*.standby\_file\_management='auto'

\*.undo\_tablespace='UNDOTBS1'

#### \*.db\_unique\_name=orclcdb

**\*.fal\_server=orclcdb**

**\*.log\_file\_name\_convert='ORCLCDB','orclcdbFS'**

1. Copy the /tmp/orclcdbFS.ctl Far Sync control file to the two destination directories used in the initialization parameter file. Rename the files to the correct name while copying them.

[oracle@host02 ~]$ **cp /tmp/orclcdbFS.ctl**

**/u01/app/oracle/oradata/orclcdbFS/orclcdbFS01.ctl**

[oracle@host02 ~]$ **cp /tmp/orclcdbFS.ctl**

**/u01/app/oracle/fast\_recovery\_area/orclcdbFS**/**orclcdbFS02**.**ctl**

1. Copy the password file staged into the /tmp directory to the default location of Far Sync instance. Rename the file appropriately while copying it.

[oracle@host02 ~]$ **cp /tmp/orapworclcdb**

**/u01/app/oracle/product/19.3.0/dbhome\_1/dbs/orapworclcdbFS**

1. Verify that the environment variables are defined for orclcdbFS. If they are not, then use the oraenv utility to set them if needed.

[oracle@host02 ~]$ **set | grep ORA**

OLD\_ORACLE\_BASE=

ORABASE\_EXEC=/u01/app/oracle/product/19.3.0/dbhome\_1/bin/orabase ORACLE\_BASE=/u01/app/oracle ORACLE\_HOME=/u01/app/oracle/product/19.3.0/dbhome\_1 ORACLE\_SID=orclcdbFS ORAHOME=/u01/app/oracle/product/19.3.0/dbhome\_1

ORASID=oracle

[oracle@host02 ~]$

1. Use SQL\*Plus to create a binary server parameter file from the text parameter file. Create the server parameter file into the default directory.

[oracle@host02 ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Wed Jun 3 15:45:55 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **create spfile from pfile='/tmp/initorclcdbFS.ora';**

File created.

1. Start up the Far Sync instance in MOUNT mode.

SQL> **startup mount**

ORACLE instance started.

Total System Global Area 517763072 bytes Fixed Size 2290216 bytes Variable Size 440405464 bytes

Database Buffers 71303168 bytes

Redo Buffers 3764224 bytes Database mounted.

1. Use a terminal window on localhost logged in as oracle with the environment variables set to orclcdb, the primary database. Launch SQL\*plus and examine the current value of log\_archive\_dest\_2.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Wed Jun 3 15:45:55 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **show parameter log\_archive\_dest\_2**

NAME TYPE VALUE

log\_archive\_dest\_2 string

SERVICE=stndby ASYNC REOPEN=15

valid\_for=(ONLINE\_LOGFILES,

PRIMARY\_ROLE)

db\_unique\_name=stndby

1. The primary is currently forwarding redo to the physical standby database. Alter the primary database to now forward redo to the Far Sync instance instead of the physical standby database. Be sure to make the corrections both in memory and written to the server parameter file.

SQL> **alter system set log\_archive\_dest\_2='SERVICE=orclcdbFS SYNC REOPEN=15 valid\_for=(ONLINE\_LOGFILES,PRIMARY\_ROLE)**

**db\_unique\_name=orclcdbFS' scope=both;**

System altered.

1. Determine the most recently archived redo log on the primary database. Perform a log switch, and verify the next sequence number used.

SQL> **SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED\_LOG GROUP BY THREAD#;**

MAX(SEQUENCE#) THREAD#

21

1

SQL> **alter system switch logfile;**

System altered.

SQL> **SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED\_LOG GROUP BY THREAD#;**

MAX(SEQUENCE#) THREAD#

22

1

1. Use the previous SQL\*Plus session for host02 logged in as oracle with the environment variables set to orclcdbFS, the Far Sync. Verify that the last sequence number of the primary was received on the Far Sync.

SQL> **SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED\_LOG GROUP BY THREAD#;**

MAX(SEQUENCE#) THREAD#

22

1

1. Open a terminal window on stndby logged in as oracle with the environment variables set to stndby. Launch SQL\*Plus and verify that the physical standby on stndby is receiving redo from the Far Sync on host02.

[oracle@stndby]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Wed Jun 3 15:45:55 2020

Version 19.3.0.0.0

(c) 1982, 2019, Oracle. All rights reserved. Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED\_LOG GROUP BY THREAD#;**

MAX(SEQUENCE#) THREAD#

22

1

1. Exit all SQL\*Plus sessions from localhost, host02, and stndby. Leave the terminal sessions open with the environment variables set for each system.

(localhost) SQL> **exit;**

(host02) SQL> **exit;**

(stndby) SQL> **exit;**

1. Verify on the Far Sync server, host02, that the standby redo logs were automatically created.

[oracle@host02 ~]$ **ls -la**

**/u01/app/oracle/oradata/orclcdbFS/stdby\***

-rw-r-----. 1 oracle oinstall 209715712 Jun 8 15:39

/u01/app/oracle/oradata/orclcdbFS/stdbyredo01.log

-rw-r-----. 1 oracle oinstall 209715712 Jun 8 15:37

/u01/app/oracle/oradata/orclcdbFS/stdbyredo02.log

-rw-r-----. 1 oracle oinstall 209715712 Jun 8 15:37

/u01/app/oracle/oradata/orclcdbFS/stdbyredo03.log

-rw-r-----. 1 oracle oinstall 209715712 Jun 8 15:38

/u01/app/oracle/oradata/orclcdbFS/stdbyredo04.log

[oracle@host02 ~]$

## Practice 6-2: Add 2nd Far Sync to the Data Guard Environment

### Overview

In this practice, you will create an additional Far Sync instance (stndbyFS) on host04 that is in close proximity to the physical standby database. This will be used in future practices that perform role reversal. With a role reversal, the stndby instance on stndby will become the primary database. It will need to forward redo to stndbyFS Far Sync on host04.

In the practice, we copied a Far Sync control file, initialization parameter file, and password file to the /tmp directory on host02. We will start with these files and copy them to host04 since they were already edited for a Far Sync environment.

### Tasks

1. Use a terminal window for host02 logged in as oracle with the environment variables set to orclcdbFS. Copy the three files staged in /tmp to host04.

[oracle@host02 ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdbFS**

The Oracle base has been set to /u01/app/oracle [oracle@host02 ~]$

[oracle@host02 ~]$ **cat /home/oracle/setup/copy\_stndbyFS.sh**

...

scp /tmp/orclcdbFS.ctl oracle@host04:/tmp

scp /tmp/initorclcdbFS.ora oracle@host04:/tmp scp /tmp/orapworclcdb oracle@host04:/tmp

[oracle@host02 ~]$ **/home/oracle/setup/copy\_stndbyFS.sh**

/home/oracle/setup/copy\_stndbyFS.sh

The authenticity of host 'host04 (192.0.2.14)' can't be established.

ECDSA key fingerprint is SHA256:JKkb1E9vmYSa8YTFMVqZGa/vXENSXcCJJehEX+UdHfA.

ECDSA key fingerprint is MD5:74:b9:98:32:37:24:52:3d:f7:a8:12:ac:38:c3:c8:94.

Are you sure you want to continue connecting (yes/no)? **yes**

Warning: Permanently added 'host04,192.0.2.14' (ECDSA) to the list of known hosts."

oracle@host04's password: *<password>*

orclcdbFS.ctl 100% 18MB 52.9MB/s 00:00 oracle@host04's password: *<password>*

initorclcdbFS.ora 100% 1851 19.0KB/s 00:00 oracle@host04's password: *<password>*

orapworclcdb 100% 2560 239.8KB/s 00:00 [oracle@host02 ~]$

1. Open a new terminal window for host04 logged in as oracle with the environment variables set to stndbyFS. Create the initial directories needed on the Far Sync server. These are the same directories that were created on the physical standby server stndby in practice 3-2.

[oracle@host04 ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndbyFS**

The Oracle base has been set to /u01/app/oracle [oracle@host04 ~]$

[oracle@host04 ~]$ **cat /home/oracle/setup/crdir\_host04.sh**

...

mkdir -p /u01/app/oracle/admin/stndbyFS/adump mkdir -p /u01/app/oracle/oradata/stndbyFS

mkdir -p /u01/app/oracle/oradata/stndbyFS/pdbseed mkdir -p /u01/app/oracle/oradata/stndbyFS/dev1

mkdir -p /u01/app/oracle/fast\_recovery\_area/stndbyFS [oracle@host04 ~]$

[oracle@host04 ~]$ /**home/oracle/setup/crdir\_host04.sh**

[oracle@host04 ~]$

**Note:** Linux directory and file names are case-sensitive. Throughout these labs, the names for Far Sync use the format orclcdbFS and stndbyFS for readability. The last two letters are in uppercase.

1. Rename the /tmp/initorclcdbFS.ora file /tmp/initstndbyFS.ora. You will need to make changes to the initialization parameters to reflect the name change from orclcdb to stndby. You will also need to make adjustments for control file name changes and directory name changes.

[oracle@host04 ~]$ **mv /tmp/initorclcdbFS.ora**

**/tmp/initstndbyFS.ora**

[oracle@host04 ~]$ **vi /tmp/initstndbyFS.ora**

* 1. Globally, search and replace all occurrences of orclcdb with stndby

**:%s/orclcdb/stndby/g**

26 substitutions on 22 lines

* 1. Three of the substitutions from the global search and replace are incorrect and need to be reverted to their original value. Continue editing the file to locate and correct db\_name, log\_archive\_config, and log\_file\_name\_convert entries. Ask your instructor for assistance if you need help with VI syntax. Remember to always use the <ESC> key before starting a new VI command option.

\*.db\_name='**orclcdb**'

\*.log\_archive\_config='dg\_config=(**orclcdb**,**orclcdbFS**,stndby,stndbyFS

,stndby2)'

\*.log\_file\_name\_convert='**ORCLCDB**,'stndbyFS'

* 1. The entry for log\_archive\_dest\_2 is defined for a standby database role to ship redo to the stndby service. If stndby becomes the primary database, then the stndbyFS Far Sync should ship redo to orclcdb, which will become the standby database after role reversal. Correct the values for this parameter. (Changes are shown in bold.)

\*.log\_archive\_dest\_2='SERVICE=**orclcdb ASYNC** REOPEN=15 valid\_for=(STANDBY\_LOGFILES,STANDBY\_ROLE) db\_unique\_name=**orclcdb**'

* 1. Recheck all modifications and case-sensitivity issues. If accurate, then save the changes made to the file. Less changes were needed since we started with a modified file.

**:wq!**

* 1. The complete edited file is listed below for reference. Bold entries indicate changes that were made.

[oracle@host04 ~]$ **cat /tmp/initstndbyFS.ora stndbyFS**. data\_transfer\_cache\_size=0 **stndbyFS**. db\_cache\_size=46137344

**stndbyFS**. java\_pool\_size=12582912 **stndbyFS**. large\_pool\_size=12582912

**stndbyFS**. oracle\_base='/u01/app/oracle'#ORACLE\_BASE set from environment

**stndbyFS**. pga\_aggregate\_target=209715200 **stndbyFS**. sga\_target=310378496

**stndbyFS**. shared\_io\_pool\_size=4194304

**stndbyFS**. shared\_pool\_size=226492416 **stndbyFS**. streams\_pool\_size=0

\*.audit\_file\_dest='/u01/app/oracle/admin/**stndbyFS**/adump'

\*.audit\_trail='db'

\*.compatible='12.1.0.0.0'

\*.control\_files='/u01/app/oracle/oradata/**stndbyFS**/**stndbyFS01**.ctl ','/u01/app/oracle/fast\_recovery\_area/**stndbyFS**/**stndbyFS02**.ctl'

\*.db\_block\_size=8192

\*.db\_domain='example.com'

\*.db\_name='**orclcdb**'

\*.db\_recovery\_file\_dest='/u01/app/oracle/fast\_recovery\_area'

\*.db\_recovery\_file\_dest\_size=10g

\*.diagnostic\_dest='/u01/app/oracle'

\*.dispatchers='(PROTOCOL=TCP) (SERVICE=**stndbyFS**XDB)'

\*.enable\_pluggable\_database=true

**#**\*.local\_listener='LISTENER\_stndbyFS'

\*.log\_archive\_config='dg\_config=(**orclcdb,orclcdbFS,stndby,stndbyFS**

#### ,stndby2)'

\*.log\_archive\_dest\_1='location=USE\_DB\_RECOVERY\_FILE\_DEST valid\_for=(ALL\_LOGFILES,ALL\_ROLES) db\_unique\_name=**stndbyFS**'

\*.log\_archive\_dest\_2='SERVICE=**orclcdb ASYNC** REOPEN=15 valid\_for=(STANDBY\_LOGFILES,STANDBY\_ROLE) db\_unique\_name=**orclcdb**'

\*.log\_archive\_dest\_state\_1='enable'

\*.log\_archive\_format='arch\_%t\_%s\_%r.log'

\*.log\_archive\_max\_processes=4

\*.memory\_target=496m

\*.open\_cursors=300

\*.processes=300

\*.remote\_login\_passwordfile='EXCLUSIVE'

\*.standby\_file\_management='auto'

\*.undo\_tablespace='UNDOTBS1'

\*.db\_unique\_name=**stndbyFS**

\*.fal\_server=**stndby**

\*.log\_file\_name\_convert='ORCLCDB','stndbyFS'

1. Copy the /tmp/orclcdbFS.ctl Far Sync control file to the two destination directories used in the initialization parameter file. Rename the files to the correct name while copying them.

[oracle@host04 ~]$ **cp /tmp/orclcdbFS.ctl**

**/u01/app/oracle/oradata/stndbyFS/stndbyFS01.ctl**

[oracle@host04 ~]$ **cp /tmp/orclcdbFS.ctl**

**/u01/app/oracle/fast\_recovery\_area/stndbyFS**/**stndbyFS02**.**ctl**

1. Copy the password file staged into the /tmp directory to the default location of Far Sync instance. Rename the file appropriately while copying it.

[oracle@host04 ~]$ **cp /tmp/orapworclcdb**

**/u01/app/oracle/product/19.3.0/dbhome\_1/dbs/orapwstndbyFS**

1. Verify that the environment variables are defined for stndbyFS. If they are not, then use the oraenv utility to set them if needed.

[oracle@host04 ~]$ **set | grep ORA**

OLD\_ORACLE\_BASE=

ORABASE\_EXEC=/u01/app/oracle/product/19.3.0/dbhome\_1/bin/orabase ORACLE\_BASE=/u01/app/oracle ORACLE\_HOME=/u01/app/oracle/product/19.3.0/dbhome\_1 ORACLE\_SID=*stndbyFS* ORAHOME=/u01/app/oracle/product/19.3.0/dbhome\_1

ORASID=oracle

1. Use SQL\*Plus to create a binary server parameter file from the text parameter file. Create the server parameter file into the default directory.

[oracle@host04 ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Wed Jun 3 20:39:10 2020

Version 19.3.0.0.0

(c) 1982, 2019, Oracle. All rights reserved. Connected to an idle instance.

SQL> **create spfile from pfile='/tmp/initstndbyFS.ora';**

File created.

1. Start up the Far Sync instance in MOUNT mode. Leave SQL\*Plus session open when done.

SQL> **startup mount**

ORACLE instance started.

Total System Global Area 517763072 bytes Fixed Size 2290216 bytes Variable Size 440405464 bytes

Database Buffers 71303168 bytes

Redo Buffers 3764224 bytes Database mounted.

SQL> **exit;**

1. Use a terminal window on stndby logged in as oracle with the environment variables set to stndby, the physical standby database. Launch SQL\*plus and examine the current value of log\_archive\_dest\_2.

[oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Mon Jun 15 01:30:23 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **show parameter log\_archive\_dest\_2**

NAME TYPE VALUE

log\_archive\_dest\_2 string

1. The log\_archive\_dest\_2 parameter of the stndby physical standby instance is not currently defined because it is a terminal destination. After role reversals, it would become the primary database. At that time, it should forward redo to the stndbyFS Far Sync. Modify the log\_archive\_dest\_2 parameter so that it forwards redo accordingly.

SQL> **alter system set log\_archive\_dest\_2='SERVICE=stndbyFS SYNC REOPEN=15 valid\_for=(ONLINE\_LOGFILES,PRIMARY\_ROLE)**

**db\_unique\_name=stndbyFS' scope=both;**

System altered.

**Note:** At this point in the labs, the stndbyFS Far Sync has been started, but it is not currently used. It will be tested after switchover exercises are performed in future labs.

1. Exit all SQL\*Plus sessions from localhost, host02, stndby, and host04 if they are still open. Leave the terminal sessions open with the environment variables set for each system.

(localhost) SQL> **exit;** (host02) SQL> **exit;** (stndby) SQL> **exit;**

(host04) SQL> **exit;**

# Practices for Lesson 7: Creating and Managing a Snapshot Standby Database

## Practices for Lesson 7: Overview

### Practices Overview

In these practices, you will convert the physical standby database to a snapshot database and open it for read-write operations. You will create new schema objects in the database to verify the success of creating the snapshot. Finally, you will convert it back into a physical standby database, discarding the schema objects that were created.

## Practice 7-1: Convert Physical Standby to a Snapshot Standby

### Overview

In this practice, you will convert the stndby physical standby database to a snapshot standby database.

### Tasks

1. Use a terminal window logged in as oracle to stndby with the environment variables set for stndby appropriately. Launch SQL\*Plus and verify that the current database role is physical standby.

[oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Wed Jun 3 20:53:36 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **select database\_role from v$database;**

DATABASE\_ROLE PHYSICAL STANDBY SQL>

1. Verify that flashback database is turned off, and show the default value for the flashback retention target.

SQL> **select flashback\_on from v$database;**

FLASHBACK\_ON NO

SQL> **show parameter DB\_FLASHBACK\_RETENTION\_TARGET**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME |  | TYPE |  | VALUE |
| db\_flashback\_retention\_target |  | integer |  | 1440 |

1. Display the values for the two initialization parameters that define the Fast Recovery Area.

SQL> **show parameter DB\_RECOVERY\_FILE\_DEST**

NAME TYPE VALUE

db\_recovery\_file\_dest

string

/u01/app/oracle/fast\_recovery\_area

db\_recovery\_file\_dest\_size big integer 15000M

1. Display the current file types, number of files for each type, and percentage of space utilization per file type for the Fast Recovery Area.

**Note:** Your output varies.

SQL> **select file\_type,number\_of\_files,percent\_space\_used from v$recovery\_area\_usage;**

AUXILIARY DATAFILE COPY

0

0

8 rows selected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FILE\_TYPE |  |  | NUMBER\_OF\_FILES |  | PERCENT\_SPACE\_USED |
| CONTROL FILE |  |  | 0 |  | 0 |
| REDO LOG |  |  | 0 |  | 0 |
| ARCHIVED LOG |  |  | 4 |  | .15 |
| BACKUP PIECE |  |  | 0 |  | 0 |
| IMAGE COPY |  |  | 0 |  | 0 |
| FLASHBACK LOG |  |  | 0 |  | 0 |
| FOREIGN ARCHIVED | LOG |  | 0 |  | 0 |

1. Attempt to convert the physical standby database to a snapshot standby database.

SQL> **alter database convert to snapshot standby;**

alter database convert to snapshot standby

\*

ERROR at line 1:

ORA-38784: Cannot create restore point 'SNAPSHOT\_STANDBY\_REQUIRED\_06/03/2020

20:50:13'.

ORA-01153: an incompatible media recovery is active

1. Cancel redo apply on the physical standby database and reattempt to convert the physical standby database to a snapshot standby database.

SQL> **alter database recover managed standby database cancel;**

Database altered.

SQL> **alter database convert to snapshot standby;**

Database altered.

1. Display the current database role.

SQL> **select database\_role from v$database;**

DATABASE\_ROLE SNAPSHOT STANDBY

1. Verify that flashback database was automatically enabled when the physical standby was converted to a snapshot standby.

SQL> **select flashback\_on from v$database;**

FLASHBACK\_ON RESTORE POINT ONLY

1. Display the name of the guaranteed restore point that was created and its current storage size.

SQL> **select name, storage\_size from v$restore\_point;**

NAME

STORAGE\_SIZE

SNAPSHOT\_STANDBY\_REQUIRED\_06/03/2020 20:56:01 209715200

1. Display the current open mode for the snapshot standby.

SQL> **select open\_mode from v$database;**

OPEN\_MODE MOUNTED

1. Verify that a flashback log was automatically created in the Recovery Area.

**Note:** Your output varies, but you should see one or more flashback logs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SQL> **select file\_type, v$recovery\_area\_usage;** | **number\_of\_files,** |  | **percent\_space\_used** | **from** |
| FILE\_TYPE | NUMBER\_OF\_FILES |  | PERCENT\_SPACE\_USED |  |
| CONTROL FILE | 0 |  | 0 |  |
| REDO LOG | 0 |  | 0 |  |
| ARCHIVED LOG | 4 |  | .15 |  |
| BACKUP PIECE | 0 |  | 0 |  |
| IMAGE COPY | 0 |  | 0 |  |
| FLASHBACK LOG | 2 |  | .98 |  |
| FOREIGN ARCHIVED LOG | 0 |  | 0 |  |
| AUXILIARY DATAFILE COPY | 0 |  | 0 |  |
| 8 rows selected. |  |  |  |  |

1. Attempt to convert the snapshot standby back to a physical standby.

SQL> **alter database convert to physical standby;**

alter database convert to physical standby

\*

ERROR at line 1:

ORA-16433: The database or pluggable database must be opened in read/write

mode.

**Note:** Even though the command in step 6 succeeded in converting the physical standby into a snapshot standby, it must be opened into read-write mode at least one time before you can reverse the operation back to a physical standby. The purpose of this example is to show what happens if you changed your mind and wanted to convert back to a physical standby without proceeding.

1. Open the snapshot database and verify that it has been opened in read-write mode.

SQL> **alter database open;**

Database altered.

SQL> **select open\_mode from v$database;**

OPEN\_MODE READ WRITE

1. Switch the container to the DEV1 PDB for the session. Even though the container is open, the DEV1 PDB is mounted. Open the DEV1 PDB, and create a miscellaneous table. Insert 1 row into the table and commit the result. Return to the root container when done.

SQL> **alter session set container = DEV1;**

Session altered.

SQL> **alter database open;**

Database altered.

SQL> **create table misc1 (x varchar2(50));**

Table created.

SQL> **insert into misc1 values ('Test Row');**

1 row created.

SQL> **commit;**

Commit complete.

SQL> **alter session set container = CDB$ROOT;**

Session altered.

1. In step 6, managed recovery was stopped for the snapshot database. Use a terminal window logged in as oracle to localhost to with the environment variables set for orclcdb appropriately. Determine the last archived redo log for the primary database and perform a log switch. Exit SQL\*Plus when done.datadat

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Wed Jun 3 21:04:42 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED\_LOG GROUP BY THREAD#;**

MAX(SEQUENCE#)

THREAD#

27

1

SQL> **alter system switch logfile;**

System altered. SQL> **exit;**

1. Return to the SQL\*Plus session on stndby and verify that the snapshot standby is still receiving redo from the primary database, forwarded to the Far Sync, and then to the snapshot standby. The sequence number should be the next one after the number displayed on the primary database in the previous step.

SQL> **SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED\_LOG GROUP BY THREAD#;**

MAX(SEQUENCE#) THREAD#

28

1

## Practice 7-2: Convert Snapshot Standby Back to Physical Standby

### Overview

In this practice, you convert the snapshot standby back into a physical standby database.

### Tasks

1. Use the terminal session connected to stndby. Attempt to convert the snapshot standby back into a physical standby.

SQL> **alter database convert to physical standby;**

alter database convert to physical standby

\*

ERROR at line 1:

ORA-01126: database must be mounted in this instance and not open in any Instance

1. Shut down the snapshot standby and start it back up in MOUNT mode.

SQL> **shutdown immediate**

Database closed. Database dismounted.

ORACLE instance shut down.

SQL> **startup mount**

ORACLE instance started.

Total System Global Area 517763072 bytes Fixed Size 2290216 bytes Variable Size 440405464 bytes

Database Buffers 71303168 bytes

Redo Buffers 3764224 bytes

Database mounted.

1. Reattempt to convert the snapshot standby back into a physical standby.

SQL> **alter database convert to physical standby;**

Database altered.

1. Verify that flashback has been turned off, and the flashback log was deleted freeing up space in the flash recovery area.

SQL> **select flashback\_on from v$database;**

FLASHBACK\_ON NO

SQL> **select file\_type,number\_of\_files,percent\_space\_used from v$recovery\_area\_usage;**

AUXILIARY DATAFILE COPY

0

0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FILE\_TYPE |  |  | NUMBER\_OF\_FILES |  | PERCENT\_SPACE\_USED |
| CONTROL FILE |  |  | 0 |  | 0 |
| REDO LOG |  |  | 0 |  | 0 |
| ARCHIVED LOG |  |  | 5 |  | .19 |
| BACKUP PIECE |  |  | 1 |  | .17 |
| IMAGE COPY |  |  | 0 |  | 0 |
| FLASHBACK LOG |  |  | 0 |  | 0 |
| FOREIGN ARCHIVED | LOG |  | 0 |  | 0 |

1. Open the container database to enable Active Data Guard, and then switch the session to the DEV1 pluggable database. Open the DEV1 PDB.

SQL> **alter database open;**

Database altered.

SQL> **alter session set container = DEV1;**

Session altered.

SQL> **alter database open;**

Database altered.

1. Attempt to query the MISC1 table that was created, and verify that the table no longer exists after converting the snapshot standby back into a physical standby database.

SQL> **select \* from misc1;**

select \* from misc1

\* ERROR at line 1:

ORA-00942: table or view does not exist

SQL> **select table\_name from dba\_tables where table\_name like 'MISC%';**

no rows selected

1. Return the session back to the container root. Shut down the physical standby to disable Active Data Guard. Start in back up in the MOUNT state and restart Redo Apply.

SQL> **alter session set container = CDB$ROOT;**

Session altered.

SQL> **shutdown immediate**

Database closed. Database dismounted.

ORACLE instance shut down.

SQL> **startup mount**

ORACLE instance started.

Total System Global Area 517763072 bytes Fixed Size 2290216 bytes Variable Size 440405464 bytes

Database Buffers 71303168 bytes

Redo Buffers 3764224 bytes Database mounted.

SQL> **alter database recover managed standby database disconnect;**

Database altered.

1. Exit SQL\*Plus on any host machine in which it is open. Do not close the terminal sessions.

SQL> **exit;**

# Practices for Lesson 8: Creating a Logical Standby Database

## Practices for Lesson 8: Overview

### Practices Overview

In these practices, you will prepare stndby to create a logical standby database. You will use the RMAN utility to create the physical standby database and then verify its operation.

## Practice 8-1: Identify Unsupported Objects for Logical Standbys

### Overview

In this practice, you will examine the primary database to determine which objects will not be supported in a logical standby.

### Tasks

1. Use a terminal window on localhost connected as oracle with the environment variables set to orclcdb. Start SQL\*Plus and verify that all pluggable databases are open for queries. If any pluggable database is mounted or shut down, the queries that follow this step will not return complete results.COL

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Thu Jun 4 09:44:37 2020

Version 19.3.0.0.0

(c) 1982, 2019, Oracle. All rights reserved. Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **col name format a30**

SQL> **select con\_id, name, open\_mode from v$containers;**

CON\_ID NAME

OPEN\_MODE

1. CDB$ROOT
2. PDB$SEED
3. DEV1

READ WRITE

READ ONLY READ WRITE

1. Find all tables across all PDBs without unique logical identifiers in the primary database. This query will take a few minutes to run.

SQL> **SELECT CON\_ID, OWNER, TABLE\_NAME FROM CDB\_LOGSTDBY\_NOT\_UNIQUE WHERE (CON\_ID, OWNER, TABLE\_NAME) NOT IN (SELECT DISTINCT CON\_ID, OWNER, TABLE\_NAME FROM CDB\_LOGSTDBY\_UNSUPPORTED) AND BAD\_COLUMN = 'Y';**

no rows selected

**Note:** This query differs from the one listed in the product documentation. It has been modified to examine schema objects across all PDBs.

1. Identify the internal schemas that ship with the Oracle Database. Any user-defined table created into these schemas will not be replicated on the logical standby database. Also, those user-defined tables will not show up in the DBA\_LOGSTDBY\_UNSUPPORTED or CDB\_LOGSTDBY\_UNSUPPORTED views of step 4, even though they are unsupported.

SQL> **col owner format a30**

SQL> **SELECT CON\_ID, OWNER FROM CDB\_LOGSTDBY\_SKIP WHERE STATEMENT\_OPT = 'INTERNAL SCHEMA' ORDER BY CON\_ID, OWNER;**

CON\_ID OWNER

...

3 WMSYS

3 XDB

3 XS$NULL

69 rows selected.

**Note:** This query differs from the one listed in the product documentation. It has been modified to examine schema objects across all PDBs.

1. Identify tables that do not belong to internal schemas and that will not be maintained by SQL Apply because of unsupported data types.

SQL> **SELECT DISTINCT CON\_ID,OWNER,TABLE\_NAME FROM CDB\_LOGSTDBY\_UNSUPPORTED ORDER BY OWNER,TABLE\_NAME;**

no rows selected

1. View the column names and data types that conflict with SQL Apply. Exit SQL\*Plus when done.

SQL> **SELECT CON\_ID, COLUMN\_NAME,DATA\_TYPE FROM CDB\_LOGSTDBY\_UNSUPPORTED;**

no rows selected SQL> **exit**

[oracle@localhost ~]$

## Practice 8-2: Create a Logical Standby (Temporarily a Physical)

### Overview

In this practice, you will prepare stndby2 to receive the logical standby database and create the logical standby database using RMAN and SQL.

### Tasks

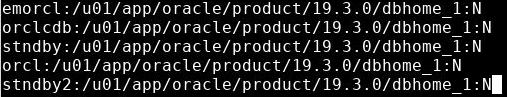
1. Use a terminal logged in as oracle to stndby2. Edit the script crdir\_stndby.sh scriptsso that it references stndby2 then Run the crdir\_stndby.sh script or manually create the dierctories required. The script will complete all preparation steps and next we will create the stndby2 standby database.

**Note:** We already performed similar tasks in practice 3-2. So, we leverage the script to

Task 1: Create Directories for stndby2 #########################################

Done #############################################

Go to /etc and edit oratab to include the stndby2 database



Task 2: Create initstndby2.ora for stndby2 #############################################

Done ###########################################

Task 3: Create password file for stndby2 ###########################################

Done ################################

Task 4: Start stndby2 NOMOUNT ################################

[oracle@8c793fb03eed]$ .oraenv

ORACLE\_SID = [stndby2]

oracle@8c793fb03eed]$ sqlplus / as sysdba

connected to an idle instance

sql> startup nomount pfile=$ORACLE\_HOME/dbs/initstndby2.ora

Task 5 Step 1 Insert stndby2 in netmgr as a service and statically add stndby2 in the listener database services.

Step 2. Bounce the listener. $ lsnrctl stop then start

Step 3.Then go to rman and connect to the target database orclcdb and auxiliary stndby2. Make sure it shows stndby2 in nomount mode.

RMAN target sys/fenago@orclcdb auxiliary sys/fenago@stndby2

Connected to target database: ORCLCDB (DBID=2860384372

Connected to auxiliary database: ORCLCDB (not mounted)

Step 4. Execute the duplicated database command below:

RMAN> duplicate target database for standby from active database ;

SQL\*Plus: Release 19.0.0.0.0 - Production on Thu Jun 4 12:50:16 2020

Version 19.3.0.0.0

(c) 1982, 2019, Oracle. All rights reserved. Connected to an idle instance.

SQL> ORACLE instance started.show

exi

Total System Global Area 268434280 bytes Fixed Size 8895336 bytes Variable Size 201326592 bytes Database Buffers 50331648 bytes

Redo Buffers 7880704 bytes

SQL> Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0 #################

Task 5: Create stndby2 STANDBY DB ####################################

1. Edit initstndby2.ora file and change all occurrences

Of stndby to stndby2.

Recovery Manager: Release 19.0.0.0.0 - Production on Thu Jun 4 12:50:24 2020

Version 19.3.0.0.0

1. connected to target database via RMAN as shown below:

RMAN target sys/fenago@orclcdb auxiliary sys/fenago@stndby2

RMAN> ORCLCDB (DBID=2732274290) connected to auxiliary database: STNDBY2 (not mounted)

1. Execute the following command to create stndby2c/

RMAN> duplicate target database for standby from active database ;

}

using target database control file instead of recovery catalog allocated channel: prmy1

channel prmy1: SID=42 device type=DISK

allocated channel: stby1

channel stby1: SID=20 device type=DISK Starting Duplicate Db at 04-JUN-20

contents of Memory Script:

{

backup as copy reuse passwordfile auxiliary format

'/u01/app/oracle/product/19.3.0/dbhome\_1/dbs/orapwstndby2' ; restore clone from service 'orclcdb' spfile to

'/u01/app/oracle/product/19.3.0/dbhome\_1/dbs/spfilestndby2.ora'

;

sql clone "alter system set spfile= ''/u01/app/oracle/product/19.3.0/dbhome\_1/dbs/spfilestndby2.ora ''";

}

executing Memory Script

Starting backup at 04-JUN-20 Finished backup at 04-JUN-20

Starting restore at 04-JUN-20

channel stby1: starting datafile backup set restore

channel stby1: using network backup set from service orclcdb channel stby1: restoring SPFILE

output file name=/u01/app/oracle/product/19.3.0/dbhome\_1/dbs/spfilestndby2. ora

channel stby1: restore complete, elapsed time: 00:00:01 Finished restore at 04-JUN-20

sql statement: alter system set spfile= ''/u01/app/oracle/product/19.3.0/dbhome\_1/dbs/spfilestndby2.ora ''

contents of Memory Script:

{

sql clone "alter system set audit\_file\_dest = ''/u01/app/oracle/admin/stndby2/adump'' comment= '''' scope=spfile";

sql clone "alter system set control\_files =

''/u01/app/oracle/oradata/stndby2/control01.ctl'', ''/u01/app/oracle/fast\_recovery\_area/stndby2/control02.ctl'' comment=

'''' scope=spfile";

sql clone "alter system set dispatchers = ''(PROTOCOL=TCP) (SERVICE=stndby2XDB)'' comment=

'''' scope=spfile";

sql clone "alter system set local\_listener = ''LISTENER\_stndby2'' comment=

'''' scope=spfile";

sql clone "alter system set db\_name = ''orclcdb'' comment=

'''' scope=spfile";

sql clone "alter system set db\_unique\_name = ''stndby2'' comment=

'''' scope=spfile";

sql clone "alter system set db\_file\_name\_convert = ''ORCLCDB'', ''stndby2'' comment=

'''' scope=spfile";

sql clone "alter system set log\_file\_name\_convert = ''ORCLCDB'', ''stndby2'' comment=

'''' scope=spfile";

sql clone "alter system set fal\_server = ''orclcdb'' comment=

'''' scope=spfile";

sql clone "alter system set log\_archive\_dest\_1 = ''location=USE\_DB\_RECOVERY\_FILE\_DEST

valid\_for=(ALL\_LOGFILES,ALL\_ROLES) db\_unique\_name=stndby2'' comment=

'''' scope=spfile"; shutdown clone immediate; startup clone nomount;

}

executing Memory Script

sql statement: alter system set audit\_file\_dest = ''/u01/app/oracle/admin/stndby2/adump'' comment= '''' scope=spfile

sql statement: alter system set control\_files = ''/u01/app/oracle/oradata/stndby2/control01.ctl'', ''/u01/app/oracle/fast\_recovery\_area/stndby2/control02.ctl'' comment= '''' scope=spfile

sql statement: alter system set dispatchers = ''(PROTOCOL=TCP) (SERVICE=stndby2XDB)'' comment= ''''

scope=spfile

sql statement: alter system set local\_listener = ''LISTENER\_stndby2'' comment= '''' scope=spfile

sql statement: alter system set db\_name = ''orclcdb'' comment= '''' scope=spfile

sql statement: alter system set db\_unique\_name = ''stndby2'' comment= '''' scope=spfile

sql statement: alter system set db\_file\_name\_convert = ''ORCLCDB'', ''stndby2'' comment= '''' scope=spfile

sql statement: alter system set log\_file\_name\_convert = ''ORCLCDB'', ''stndby2'' comment= '''' scope=spfile

sql statement: alter system set fal\_server = ''orclcdb'' comment= '''' scope=spfile

sql statement: alter system set log\_archive\_dest\_1 = ''location=USE\_DB\_RECOVERY\_FILE\_DEST valid\_for=(ALL\_LOGFILES,ALL\_ROLES) db\_unique\_name=stndby2'' comment= '''' scope=spfile

Oracle instance shut down

connected to auxiliary database (not started) Oracle instance started

Total System Global Area 629145352 bytes

Fixed Size 9137928 bytes Variable Size 188743680 bytes

Database Buffers 423624704 bytes Redo Buffers 7639040 bytes allocated channel: stby1

channel stby1: SID=255 device type=DISK

contents of Memory Script:

{

restore clone from service 'orclcdb' standby controlfile;

}

executing Memory Script Starting restore at 04-JUN-20

channel stby1: starting datafile backup set restore

channel stby1: using network backup set from service orclcdb channel stby1: restoring control file

channel stby1: restore complete, elapsed time: 00:00:02

output file name=/u01/app/oracle/oradata/stndby2/control01.ctl

output file name=/u01/app/oracle/fast\_recovery\_area/stndby2/control02.ctl

Finished restore at 04-JUN-20

contents of Memory Script:

{

sql clone 'alter database mount standby database';

}

executing Memory Script

sql statement: alter database mount standby database contents of Memory Script:

{

set newname for tempfile 1 to "/u01/app/oracle/oradata/stndby2/temp01.dbf";

set newname for tempfile 2 to

"/u01/app/oracle/oradata/stndby2/pdbseed/temp012020-06-02\_14- 25-16-052-PM.dbf";

set newname for tempfile 3 to "/u01/app/oracle/oradata/stndby2/dev1/temp01.dbf";

switch clone tempfile all; set newname for datafile 1 to

"/u01/app/oracle/oradata/stndby2/system01.dbf"; set newname for datafile 3 to

"/u01/app/oracle/oradata/stndby2/sysaux01.dbf"; set newname for datafile 4 to

"/u01/app/oracle/oradata/stndby2/undotbs01.dbf"; set newname for datafile 5 to

"/u01/app/oracle/oradata/stndby2/pdbseed/system01.dbf"; set newname for datafile 6 to

"/u01/app/oracle/oradata/stndby2/pdbseed/sysaux01.dbf"; set newname for datafile 7 to

"/u01/app/oracle/oradata/stndby2/users01.dbf"; set newname for datafile 8 to

"/u01/app/oracle/oradata/stndby2/pdbseed/undotbs01.dbf"; set newname for datafile 9 to

"/u01/app/oracle/oradata/stndby2/dev1/system01.dbf"; set newname for datafile 10 to

"/u01/app/oracle/oradata/stndby2/dev1/sysaux01.dbf"; set newname for datafile 11 to

"/u01/app/oracle/oradata/stndby2/dev1/undotbs01.dbf"; set newname for datafile 12 to

"/u01/app/oracle/oradata/stndby2/dev1/users01.dbf"; restore

from nonsparse from service 'orclcdb' clone database

;

sql 'alter system archive log current';

}

executing Memory Script executing command: SET NEWNAME executing command: SET NEWNAME executing command: SET NEWNAME

renamed tempfile 1 to

/u01/app/oracle/oradata/stndby2/temp01.dbf in control file

renamed tempfile 2 to

/u01/app/oracle/oradata/stndby2/pdbseed/temp012020-06-02\_14-25- 16-052-PM.dbf in control file

renamed tempfile 3 to

/u01/app/oracle/oradata/stndby2/dev1/temp01.dbf in control file

executing command: SET NEWNAME executing command: SET NEWNAME executing command: SET NEWNAME executing command: SET NEWNAME executing command: SET NEWNAME executing command: SET NEWNAME executing command: SET NEWNAME executing command: SET NEWNAME executing command: SET NEWNAME executing command: SET NEWNAME executing command: SET NEWNAME Starting restore at 04-JUN-20

channel stby1: starting datafile backup set restore channel stby1: using network backup set from service orclcdb

channel stby1: specifying datafile(s) to restore from backup set

channel stby1: restoring datafile 00001 to

/u01/app/oracle/oradata/stndby2/system01.dbf

channel stby1: restore complete, elapsed time: 00:00:38 channel stby1: starting datafile backup set restore

channel stby1: using network backup set from service orclcdb

channel stby1: specifying datafile(s) to restore from backup set

channel stby1: restoring datafile 00003 to

/u01/app/oracle/oradata/stndby2/sysaux01.dbf

channel stby1: restore complete, elapsed time: 00:00:25 channel stby1: starting datafile backup set restore

channel stby1: using network backup set from service orclcdb

channel stby1: specifying datafile(s) to restore from backup set

channel stby1: restoring datafile 00004 to

/u01/app/oracle/oradata/stndby2/undotbs01.dbf

channel stby1: restore complete, elapsed time: 00:00:07 channel stby1: starting datafile backup set restore

channel stby1: using network backup set from service orclcdb

channel stby1: specifying datafile(s) to restore from backup set

channel stby1: restoring datafile 00005 to

/u01/app/oracle/oradata/stndby2/pdbseed/system01.dbf channel stby1: restore complete, elapsed time: 00:00:15 channel stby1: starting datafile backup set restore channel stby1: using network backup set from service orclcdb

channel stby1: specifying datafile(s) to restore from backup set

channel stby1: restoring datafile 00006 to

/u01/app/oracle/oradata/stndby2/pdbseed/sysaux01.dbf channel stby1: restore complete, elapsed time: 00:00:07 channel stby1: starting datafile backup set restore

channel stby1: using network backup set from service orclcdb

channel stby1: specifying datafile(s) to restore from backup set

channel stby1: restoring datafile 00007 to

/u01/app/oracle/oradata/stndby2/users01.dbf

channel stby1: restore complete, elapsed time: 00:00:02 channel stby1: starting datafile backup set restore

channel stby1: using network backup set from service orclcdb

channel stby1: specifying datafile(s) to restore from backup set

channel stby1: restoring datafile 00008 to

/u01/app/oracle/oradata/stndby2/pdbseed/undotbs01.dbf channel stby1: restore complete, elapsed time: 00:00:03 channel stby1: starting datafile backup set restore

channel stby1: using network backup set from service orclcdb

channel stby1: specifying datafile(s) to restore from backup set

channel stby1: restoring datafile 00009 to

/u01/app/oracle/oradata/stndby2/dev1/system01.dbf channel stby1: restore complete, elapsed time: 00:00:15 channel stby1: starting datafile backup set restore

channel stby1: using network backup set from service orclcdb

channel stby1: specifying datafile(s) to restore from backup set

channel stby1: restoring datafile 00010 to

/u01/app/oracle/oradata/stndby2/dev1/sysaux01.dbf channel stby1: restore complete, elapsed time: 00:00:15 channel stby1: starting datafile backup set restore

channel stby1: using network backup set from service orclcdb

channel stby1: specifying datafile(s) to restore from backup set

channel stby1: restoring datafile 00011 to

/u01/app/oracle/oradata/stndby2/dev1/undotbs01.dbf channel stby1: restore complete, elapsed time: 00:00:04 channel stby1: starting datafile backup set restore

channel stby1: using network backup set from service orclcdb

channel stby1: specifying datafile(s) to restore from backup set

channel stby1: restoring datafile 00012 to

/u01/app/oracle/oradata/stndby2/dev1/users01.dbf channel stby1: restore complete, elapsed time: 00:00:01 Finished restore at 04-JUN-20

sql statement: alter system archive log current

contents of Memory Script:

{

switch clone datafile all;

}

executing Memory Script

datafile 1 switched to datafile copy

input datafile copy RECID=5 STAMP=1042203295 file name=/u01/app/oracle/oradata/stndby2/system01.dbf

datafile 3 switched to datafile copy

input datafile copy RECID=6 STAMP=1042203295 file name=/u01/app/oracle/oradata/stndby2/sysaux01.dbf

datafile 4 switched to datafile copy

input datafile copy RECID=7 STAMP=1042203295 file name=/u01/app/oracle/oradata/stndby2/undotbs01.dbf

datafile 5 switched to datafile copy

input datafile copy RECID=8 STAMP=1042203295 file name=/u01/app/oracle/oradata/stndby2/pdbseed/system01.dbf

datafile 6 switched to datafile copy

input datafile copy RECID=9 STAMP=1042203295 file name=/u01/app/oracle/oradata/stndby2/pdbseed/sysaux01.dbf

datafile 7 switched to datafile copy

input datafile copy RECID=10 STAMP=1042203295 file name=/u01/app/oracle/oradata/stndby2/users01.dbf

datafile 8 switched to datafile copy

input datafile copy RECID=11 STAMP=1042203295 file name=/u01/app/oracle/oradata/stndby2/pdbseed/undotbs01.dbf

datafile 9 switched to datafile copy

input datafile copy RECID=12 STAMP=1042203295 file name=/u01/app/oracle/oradata/stndby2/dev1/system01.dbf

datafile 10 switched to datafile copy

input datafile copy RECID=13 STAMP=1042203295 file name=/u01/app/oracle/oradata/stndby2/dev1/sysaux01.dbf

datafile 11 switched to datafile copy

input datafile copy RECID=14 STAMP=1042203295 file name=/u01/app/oracle/oradata/stndby2/dev1/undotbs01.dbf

datafile 12 switched to datafile copy

input datafile copy RECID=15 STAMP=1042203295 file name=/u01/app/oracle/oradata/stndby2/dev1/users01.dbf

Finished Duplicate Db at 04-JUN-20

allocated channel: stby

channel stby: SID=27 device type=DISK

sql statement: alter database recover managed standby database disconnect

released channel: prmy1 released channel: stby1 released channel: stby

Recovery Manager complete.

Done #################################################

Completed All 5 Tasks. Verify Your Environment #################################################

1. 1. Start the database: SQL> STARTUP NOMOUNT;
2. 2. Mount the standby database: SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
3. 3. Start the managed recovery operation: SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION;

## Practice 8-3: Start Redo Transport and Verify Operation

### Overview

In this practice, you will start the redo transport from localhost to stndby for the new physical standby and verify operation.

### Tasks

1. Use a terminal window for localhost logged in as oracle with the environment variables set to orclcdb and start redo transport by defining log\_archive\_dest\_3 pointing to the logical standby database.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Thu Jun 4 12:59:16 2020

Version 19.3.0.0.0

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

Check to make sure your log\_archive\_config parameter is set

To below. **Sql> show parameter log\_archive\_config**

SQL> alter system set log\_archive\_config=

'dg\_config=(orclcdb, stndby, orcl, stndby2)';

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter system set log\_archive\_dest\_3='SERVICE="stndby2" SYNC REOPEN=15 valid\_for=(ONLINE\_LOGFILES,PRIMARY\_ROLE)**

**db\_unique\_name="stndby2"' scope=both;**

System altered.

**Note:** For this step, we are configuring redo transportation from the primary database to the logical standby database This is designed to illustrate a typical configuration where the primary database transports redo directly to the standby site, and provide a little variation in the architecture. Again, this is for illustration only. At a later time, we will change this to use the far sync instance.

1. Determine the last sequence number archived on the primary database.

SQL> **SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED\_LOG GROUP BY THREAD#;**

MAX(SEQUENCE#) THREAD#

38

1

1. Use a terminal window on stndby2

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby2**

The Oracle base has been set to /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Thu Jun 4 13:10:26 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED\_LOG GROUP BY THREAD#;**

MAX(SEQUENCE#)

THREAD#

38

1

1. Altesta
2. connected as oracle with the environment variables set to stndby2. Start SQL\*Plus and determine the last sequence number of the physical standby instance.
3. Return to the terminal window of localhost, and force a log switch to advance the online redo log sequence number. Verify that the sequence number has increased.

SQL> **alter system switch logfile;**

System altered.

SQL> **SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED\_LOG GROUP BY THREAD#;**

MAX(SEQUENCE#) THREAD#

39

1

1. Return to the terminal window of stndby, and verify that the stndby physical standby instance is receiving redo from the primary database instance.

SQL> **SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED\_LOG GROUP BY THREAD#;**

MAX(SEQUENCE#) THREAD#

39

1

## Practice 8-4: Convert Physical Standby to Logical Standby

### Overview

In this practice, you will convert the newly created physical standby 'stndby2' to a logical standby database.

### Tasks

1. Use a terminal window on localhost connected as oracle with the environment variables set to stndby2. Stop redo apply on the stndby2 physical standby.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby2**

The Oracle base remains unchanged with value /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sat Jun 6 13:13:37 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **c;**

Database altered.

1. Use a terminal window on localhost connected as oracle with the environment variables set to orclcdb. Build the LogMiner dictionary into the redo. Wait for this procedure to finish before continuing with labs.

SQL> **execute dbms\_logstdby.build**

PL/SQL procedure successfully completed.

1. Use a terminal window on stndby2

SQL> **alter database recover to logical standby stndby2;**

Database altered.

1. connected as oracle with the environment variables set to stndby2. Continue applying redo data to the physical standby until it is ready to convert to a logical standby database.
2. Increase the SGA size allocated to the logical standidby database. **DON"T DO THIS.**

Test sga\_max\_size before using max\_target\_size.

1. Shut down the logical standby database on stndby2, and restart it in MOUNT mode.

|  |  |  |
| --- | --- | --- |
| SQL> **shutdown**  ORA-01507: database not mounted  ORACLE instance shut down. SQL> **startup mount**  ORACLE instance started.  Total System Global Area 880802384 bytes | | |
| Fixed Size | 9140816 | bytes |
| Variable Size | 440401920 | bytes |
| Database Buffers | 423624704 | bytes |
| Redo Buffers | 7634944 | bytes |
| Database mounted. |  |  |
| SQL> |  |  |

1. Display the LOG\_ARCHIVE\_DEST parameters on stndby2 that were copied from the alter sprimary database. Only entries that have values are displayed below.

SQL> **show parameter log\_archive\_dest**

NAME TYPE VALUE

log\_archive\_dest\_1 string

log\_archive\_dest\_2 string

location=USE\_DB\_RECOVERY\_FILE\_ DEST valid\_for = (ALL\_LOGFILES, ALL\_ROLES) db\_unique\_name = stndby2

SERVICE=orclcdb SYNC REOPEN=

15 valid\_for=(ONLINE\_LOGFILES, PRIMARY\_ROLE) db\_unique\_name=

orclcdb

1. Remove the LOG\_ARCHIVE\_DEST\_2 entry on stndby because this logical database will not be a target for role reversal in this course.

SQL> **alter system set log\_archive\_dest\_2='' scope=both;**

System altered.

1. Open the logical standby database. Stndby2

SQL> **alter database open resetlogs;**

Database altered.

1. Start SQL Apply to begin applying redo data that is received from the primary database.

SQL> **alter database start logical standby apply immediate;**

Database altered.

1. Open the DEV1 PDB and verify the mode that it was opened with.

SQL> **alter pluggable database dev1 open;**

Pluggable database altered. SQL> **col name format a20**

SQL> **select con\_id, name, open\_mode from v$containers;**

CON\_ID NAME

OPEN\_MODE

1. CDB$ROOT
2. PDB$SEED
3. DEV1

READ WRITE READ ONLY

READ WRITE

1. Exit SQL\*Plus sessions on all host machines. Leave the terminal session windows open with the environment variables set.

(localhost) SQL> **exit;**

(stndby) SQL> **exit;**

# Practices for Lesson 9: Oracle Data Guard Broker: Overview

## Practices for Lesson 9

There are no practices for this lesson.

# Practices for Lesson 10: Creating a Data Guard Broker Configuration

## Practices for Lesson 10: Overview

### Practices Overview

In these practices, you will examine the differences between local and remote connections to the Oracle Database instance by using the DGMGRL utility. You will also create and enable a Data Guard broker configuration.

## Practice 10-1: Establishing Local and Remote Connections with DGMGRL

### Overview

In this practice, you will use DGMGRL and connect with both local and remote connections. The password file will be updated on the primary database and copied to every other destination in the Data Guard configuration.

### Tasks DON"T DO THIS. DGMGRL is already configured. STOP!!!!

1. Use a terminal window on localhost connected as oracle with the environment variables set to orclcdb. Issue the "ID" command and verify that the dgdba operating system group is assigned to the oracle account.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@localhost]$ **id**

uid=54321(oracle) gid=54321(oinstall) groups=54321(oinstall),54322(dba),54323(oper),54324(backupdba),5 4325(dgdba),54326(kmdba)

**Note:** The oracle user is a member of the dgdba group. As part of the class setup, this group was associated with the SYSDG privilege for Data Guard.

**The sysdg grant looks like this. ex**

SQL> grant sysdg to system

as hostLaunch the DGMGRL utility and verify that you are able to connect as the SYSDG user with operating system authentication. (The oracle OS exuser is in the dgdba group.)

[oracle@localhost ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Thu Jun 4 14:36:27 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg**

Password: << sysexit >> Connected to "orclcdb"

Connected as SYSDG.

DGMGRL>

**Note:** With Operating System authentication, any password will work for local connections. However, during switchover and failover operation to the remote site, you must use the correct password.

1. Attempt to make a remote connection to the physical standby database stndby as the

SYSDG user. You must use the password that is in the password file. Exit DGMGRL.

DGMGRL> **connect sysdg@stndby**

Password: **fenago**

ORA-01017: invalid username/password; logon denied

DGMGRL> **exit**

[oracle@localhost ~]$

**Note:** During the creation of the database, the option to use the same password for all connadministrative accounts was chosen. However, this applied only to the SYS and SYSTEM database accounts.

1. Use SQL\*Plus on localhost connected as SYSDBA to reset the SYSDG password and unlock the account. Exit SQL\*Plus.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Thu Jun 4 14:39:05 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter user sysdg identified by fenago;**

User altered.

SQL> **alter user sysdg account unlock;**

User altered.

SQL> **grant sysdg to sysdg container=all;**

Grant succeeded.

SQL> alter pluggable database open all;

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

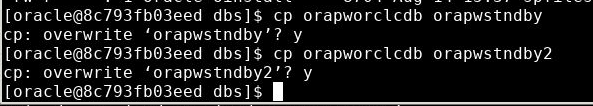
19.0.0.0.0 - Production

hostconn

Version 19.3.0.0.0

[oracle@localhost ~]$

1. In the current terminal window on localhost, copy the modified password file to all other machines overwriting the password files that are already there. On stndby, a password file is needed for both the physical standby database and the logical standby database. The password files should be renamed during the copy to the appropriate names for each destination.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | | |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Note:** As of Oracle Database 19c Release 2 (19.3.0.1), password file changes done on the primary database are automatically propagated to standby databases. The only exception to this is far sync instances. The updated password files must still be manually copied to far sync instances because far sync instances receive redo, but do not apply it.

id

1. Launch the DGMGRL utility on localhost and verify that you are now able to establish a remote connection as sysdg to the physical standby database. Exit DGMGRL when conn

[oracle@localhost ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Thu Jun 4 14:46:32 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg/<password>@stndby** Connected to "stndby"

Connected as SYSDG. DGMGRL> **exit**

## Practice 10-2: Create and Enable a Data Guard Broker Configuration

### Overview

In this practice, you will create and name the Data Guard configuration. The physical standby database, far sync instances, and logical standby database will be added to the configuration. You will enable the configuration and define redo routing rules.

### Tasks

1. Use a terminal window on localhost(primary) connected as oracle with the environment variables set to orclcdb. Connect to the primary database using SQL\*Plus and reset the LOG\_ARCHIVE\_DEST\_2 and LOG\_ARCHIVE\_DEST\_3 parameters since they are defined as network locations. Start the Data Guard Broker process. Make sure the changes are persistent. Exit SQL\*Plus.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Thu Jun 4 15:18:19 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter system set dg\_broker\_start=true scope=both;**

System altered.

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production

Version 19.3.0.0.0

[oracle@localhost ~]$

CAN"t do step 2 - No host02

1. Use a terminal window on host02 connected as oracle with the environment variables set to orclcdbFS. Connect to Far Sync using SQL\*Plus and reset the LOG\_ARCHIVE\_DEST\_2 parameter since it is defined as network location. Start the Data Guard broker process for Far Sync. Exit SQL\*Plusa

[oracle@host02 ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Thu Jun 4 15:20:05 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter system set log\_archive\_dest\_2='' scope=both;**

System altered.

SQL> **alter system set dg\_broker\_start=true scope=both;**

System altered.

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@host02 ~]$

1. Use a terminal window on stndby connected as oracle with the environment variables set to stndby. Connect to the physical standby using SQL\*Plus and stop managed recovery. Reset the LOG\_ARCHIVE\_DEST\_2 parameter because it is defined as network location. Start the Data Guard broker process for the physical standby database.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [stndby2] ? **stndby**

The Oracle base remains unchanged with value /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Thu Jun 4 15:21:52 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter database recover managed standby database cancel;**

Database altered.

Do this only if log\_archive\_dest\_2 has a value.

SQL> **alter system set log\_archive\_dest\_2='' scope=both;**

System altered.

SQL> **alter system set dg\_broker\_start=true scope=both;**

System altered.

1. If you are using a separate window for the stndby/stndby2 combination, then you can skip the part about resetting the environment variables. Otherwise, while still using the terminal window for stndby, exit SQL\*Plus. Change the environment variables to the stndby2 logical standby database. The logical standby does not have any network locations defined for redo transportation. Start the Data Guard broker process. Exit SQL\*Plus.

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0 [oracle@stndby ~]$ . **oraenv**

ORACLE\_SID = [oracle] ? **stndby2**

The Oracle base remains unchanged with value /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Thu Jun 4 15:24:21 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter database stop logical standby apply;**

Database altered.

SQL> **alter system set dg\_broker\_start=true scope=both;**

System altered.

SQL> alter pluggable database open all;

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@stndby ~]$

1. Use a terminal window on localhost connected as oracle with the environment variables set to orclcdb. Launch DGMGRL and attempt to show the configuration.

[oracle@localhost ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Thu Jun 4 15:27:39 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information.

DGMGRL> **connect sysdg/<password>@orclcdb**

Connected to "orclcdb" Connected as SYSDG. DGMGRL> **show configuration**

ORA-16532: Oracle Data Guard broker configuration does not exist

Configuration details cannot be determined by DGMGRL DGMGRL>

1. Create the Data Guard broker configuration and then show the configuration.

\*\*\*BEFORE YOU DO THE CREATE first check to see if it is already configured

DGMGRL> show configuration – if not do below

DGMGRL> **create configuration 'DRSolution' as primary database is 'orclcdb' connect identifier is orclcdb;**

Configuration "DRSolution" created with primary database "orclcdb"

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

orclcdb - Primary database Fast-Start Failover: Disabled

Configuration Status: DISABLED

**Note:** Because the Data Guard broker is a distributed framework, the DGMGRL utility can be launched from any host machine that participates in the Data Guard configuration. The labs will continue to display the machine name (localhost), and therefore, the terminal session window being used, for which the DGMGRL utility was launched. It would be acceptable though to launch it from another terminal session connected to another virtual host machine.

1. Add the physical standby database stndby to the configuration and show the results.

Before doing the below check to see if the stndby database is already in the DRSolution configuration:

DGMGRL> show configuration – If not do below.

DGMGRL> **add database 'stndby' as connect identifier is stndby;**

Database "stndby" added DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby - Physical standby database Fast-Start Failover: Disabled

Configuration Status:

DISABLED

1. Add the logical standby database stndby2 to the configuration and show the results.

AGAIN do: DGMGRL> show configuration to see if it is already added.

DGMGRL> **add database 'stndby2' as connect identifier is stndby2;**

Database "stndby2" added DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby - Physical standby database stndby2 - Logical standby database

Fast-Start Failover: Disabled

Configuration Status: DISABLED

1. Enable the Data Guard broker configuration and show the results.

DGMGRL> **enable configuration**

Enabled.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Databases:

orclcdb - Primary database orclcdbFS - Far Sync (inactive)

stndby - Physical standby database stndby2 - Logical standby database stndbyFS - Far Sync (inactive)

Fast-Start Failover: DISABLED

DISABLE database stndby2

Configuration Status: SUCCESS

DO NOT PERFORM ANY STEPS WHICH USE FAR SYNC

1. Define redo routing rules for the configuration and show the results. The current primary database orclcdb should forward redo to the Far Sync orclcdbFS synchronously. Additional redo routing rules should be created for role reversal. After role reversal, the primary database will be stndby and should forward redo to the Far Sync stndbyFS synchronously. The Far Sync stndbyFS should then forward redo to the physical standby orclcdb and the logical standby. DO NOT PERFORM. These are examples only.

DGMGRL> **EDIT DATABASE 'orclcdb' SET PROPERTY 'RedoRoutes' =**

**'(orclcdb:stndby SYNC)';**

Property "RedoRoutes" updated

DGMGRL> **EDIT 'orclcdb' SET PROPERTY 'RedoRoutes' =**

**'(orclcdb:stndby,stndby2 ASYNC)';**

Property "RedoRoutes" updated

DGMGRL> **EDIT FAR\_SYNC 'stndbyFS' SET PROPERTY 'RedoRoutes' =**

**'(stndby:orclcdb,stndby2 ASYNC)';**

Property "RedoRoutes" updated

.

Do not be concerned about warnings on standby logs because they do not match in numbers.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby - Physical standby database stndby2 - Logical standby database

Members Not Receiving Redo: stndbyFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:

SUCCESS (status updated 39 second ago)

DGMGRL>

**Note:** If your output does not match the above, do not proceed with labs until all issues have been resolved. You may need to reissue the SHOW CONFIGURATION command several times to give the Virtual Machines time to catch up with all the background operations that need to be performed. For example, in one test case, it was noted that the stndby2 logical standby database was receiving "ORA-16810: multiple errors or warnings detected for this database." To further diagnose the problem, issue the command "show database stndby2". SQL Apply had stopped with an "ORA-16768: SQL Apply is stopped" message, followed by "ORA-01304: subordinate process error. Check alert and trace logs." An examination of the alert log indicated that SQL Apply had stopped due to an "ORA-4031: unable to allocate XXX bytes of shared memory." SQL Apply was restarted with the command "edit database stndby set state='APPLY-ON'", at which time the configuration reported everything acceptable. Please consult with your instructor if you need to troubleshoot any issues.

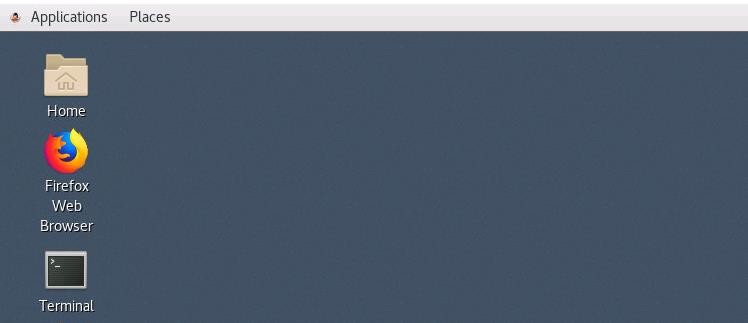
## Practice 10-3: Verify and Examine the Data Guard Environment

### Overview

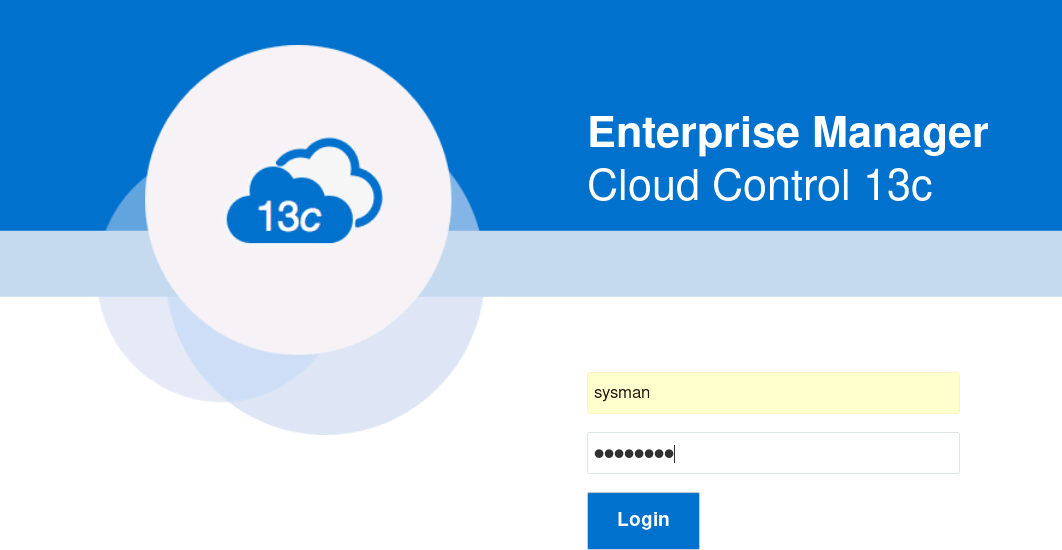
In this practice, you will discover the members of the Data Guard broker configuration, and examine and verify the Data Guard broker configuration through Enterprise Manager Cloud Control 13c.

### Tasks

1. Using the Firefox Web Browser icon, start Firefox to access to Enterprise Manager 13c.

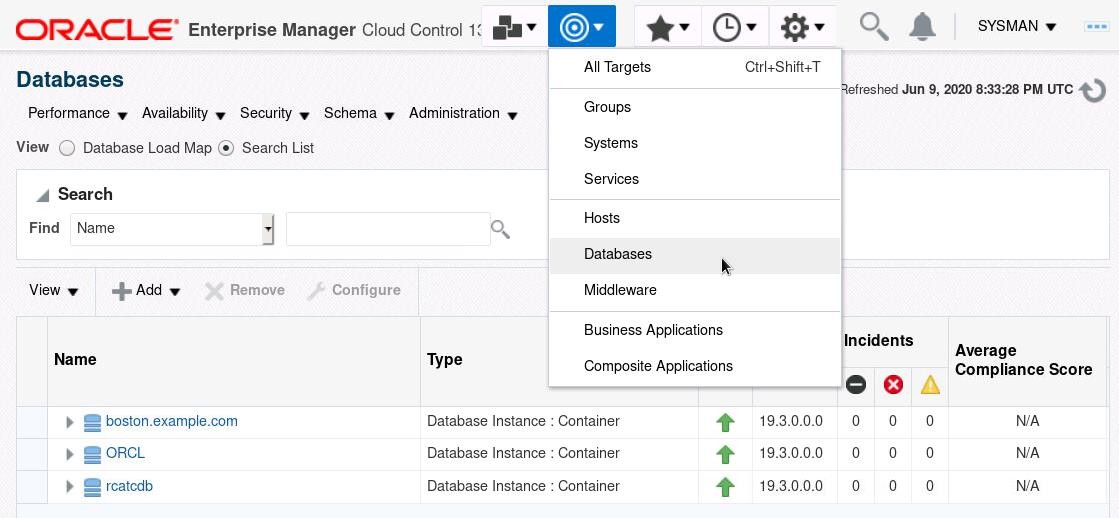


1. Enter the URL for Cloud Control. In the current setup, use
2. 
3. Log in to Enterprise Manager Cloud Control 13*c*. Log in to the application with **sysman** as the username and *<password>* as the password. The password is case-sensitive.

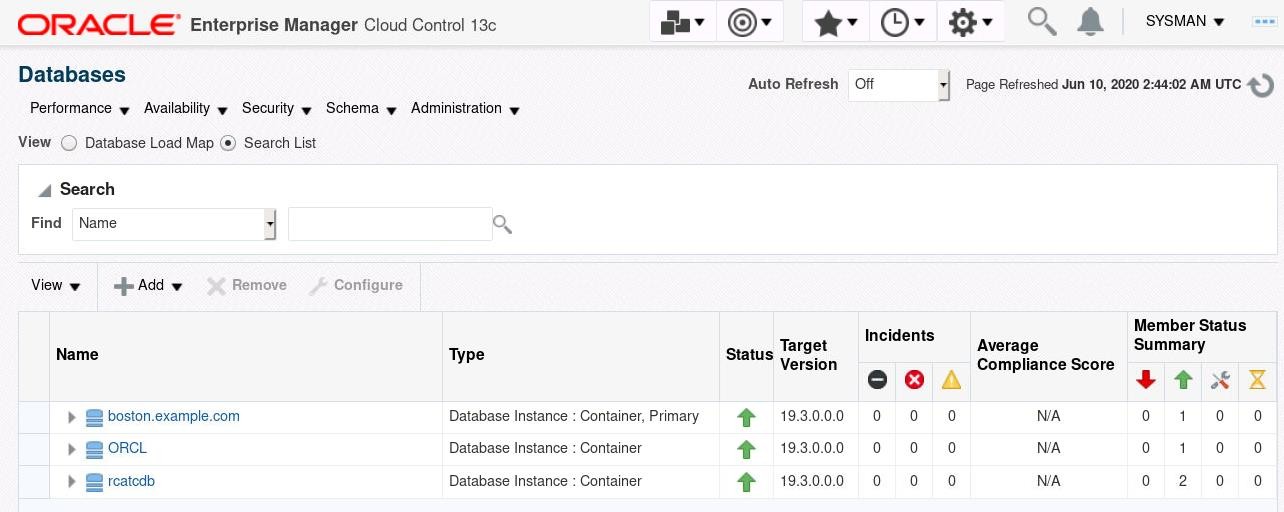


1. If you are not on the Databases page, navigate to the Databases page by selecting

**Targets**, and then **Databases** from the drop-down menu that appears.

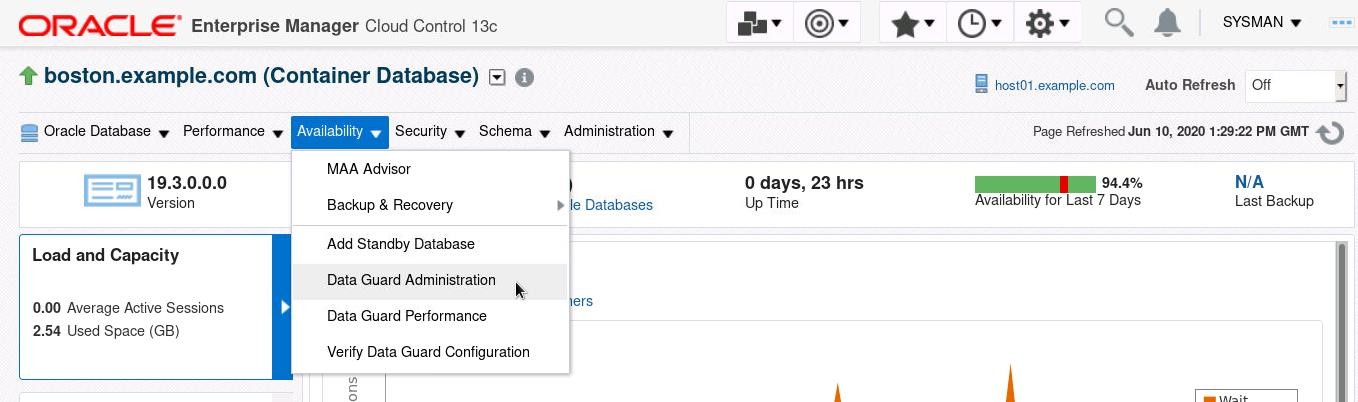


1. On the Databases page, click the link for the orclcdb target.

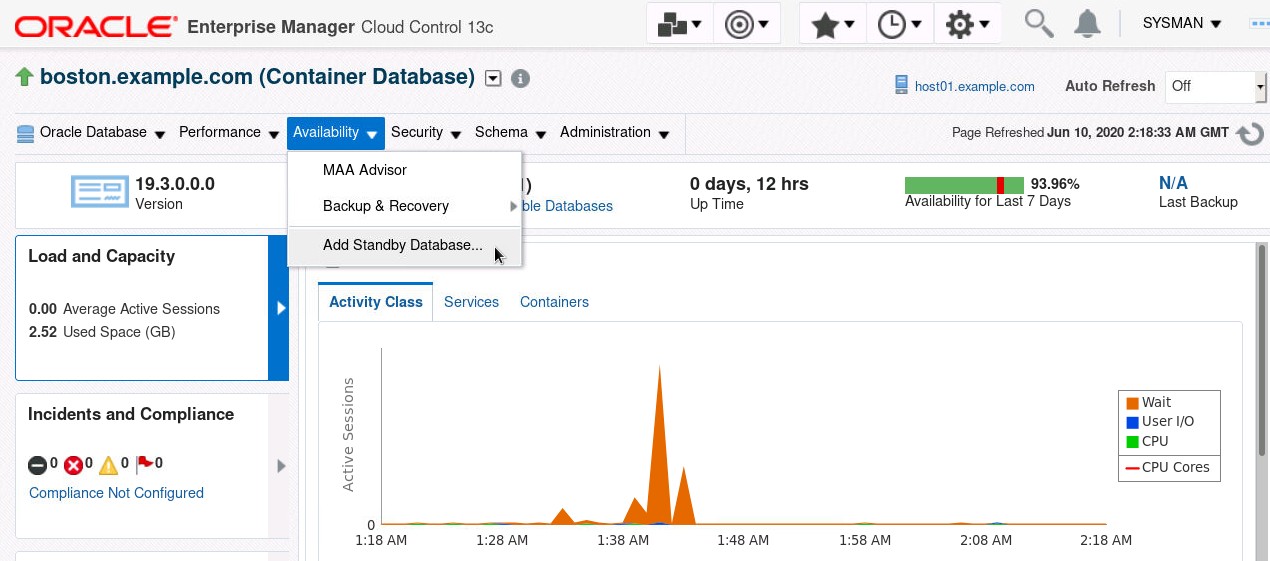


1. On the orclcdb database home page, select **Data Guard Administration**

from the Availability menu.

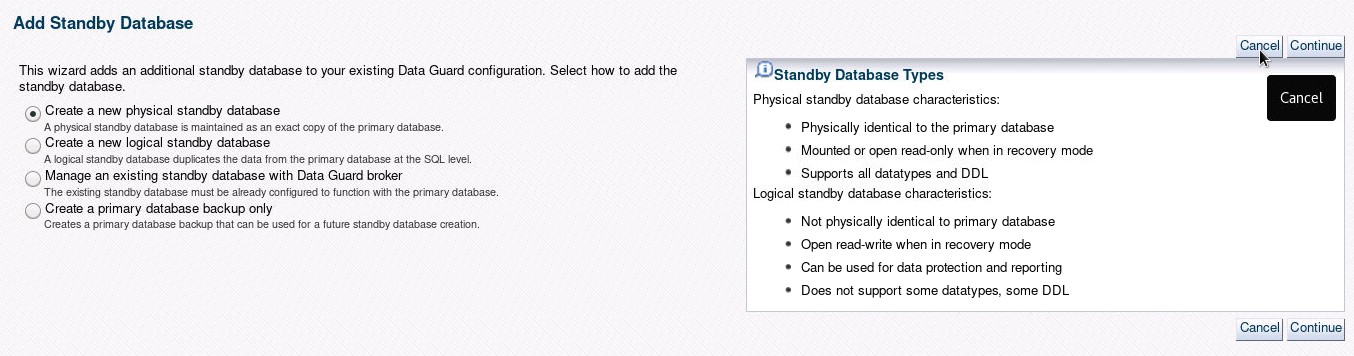


**(Optional)** If only the Add Standby Database link is visible, then select it. It will not launch the Add Standby Database Wizard, but instead will navigate to the Data Guard home page.

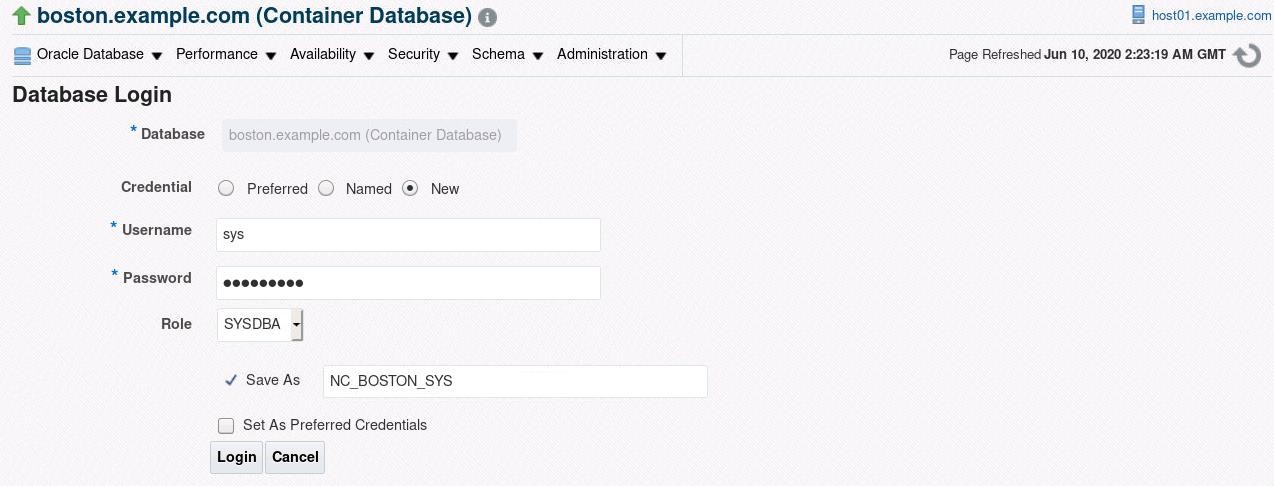


**(Optional)** If the Add Standby Database link shows the Add Standby Database page, click

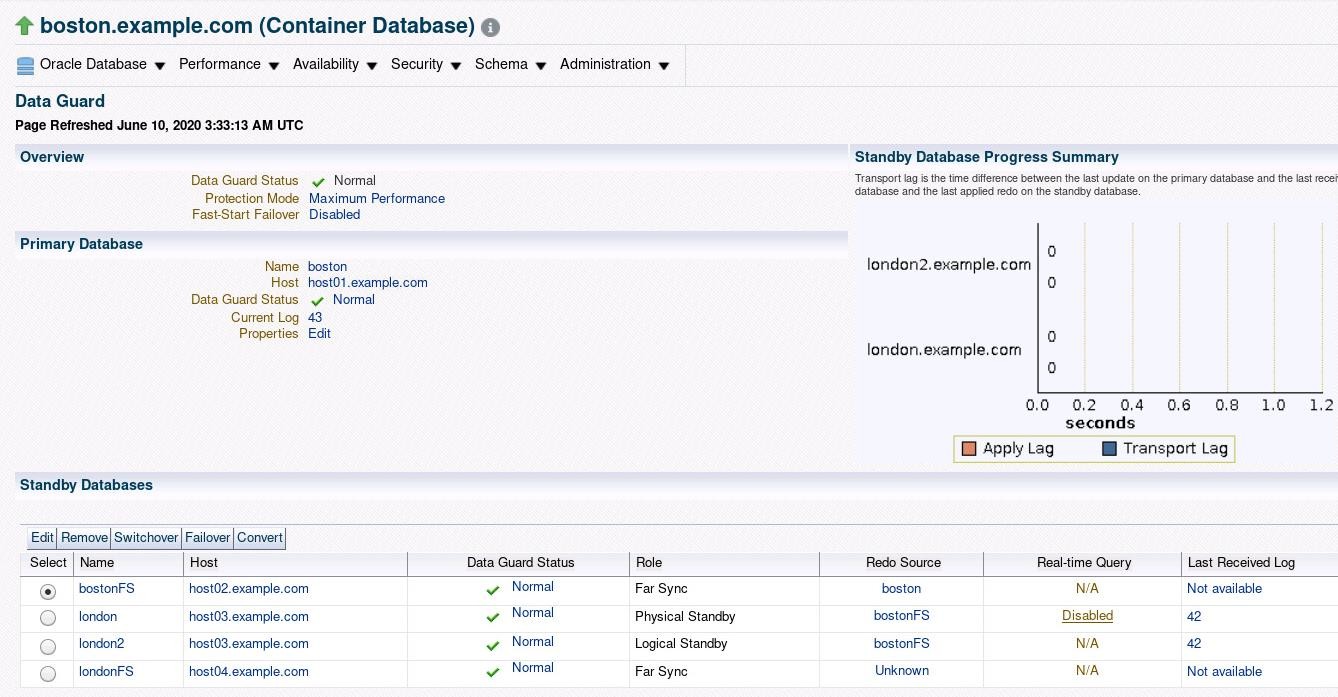
**Cancel** to navigate to the Data Guard home page.



1. On the Database Login Page, select **New** in the Credential option with the following values. Click **Login**.
   * Username: sys
   * Password: *<password>*
   * Role: SYSDBA
   * Save As: NC\_ORCLCDB\_SYS



1. The Data Guard home page, you will see the data guard configuration that you created in practice 10-2.



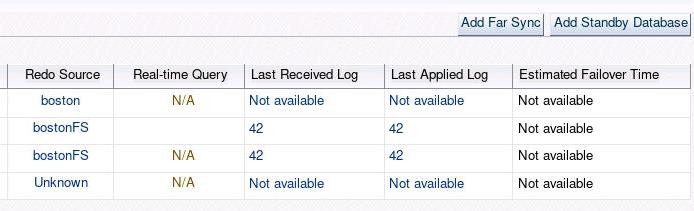
1. Review the **Overview** and **Primary Database** sections. It shows the status of the Data Guard configuration and the primary database.



1. Review the **Standby Databases** section. It shows the members of the Data Guard configuration and role/status of each member.



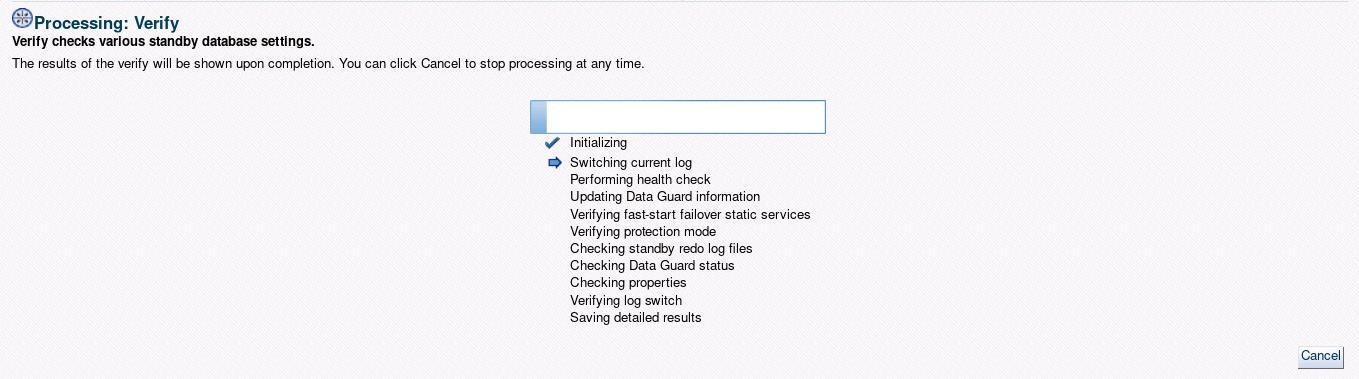
1. Continue to review the **Standby Databases** section. It shows the Redo Source of each member in the Data Guard configuration and additional information. In this section, you can also Add Far Sync or Add Standby Database.



1. Review the **Performance** and **Additional Administration** sections. It shows the additional links for the Data Guard performance.

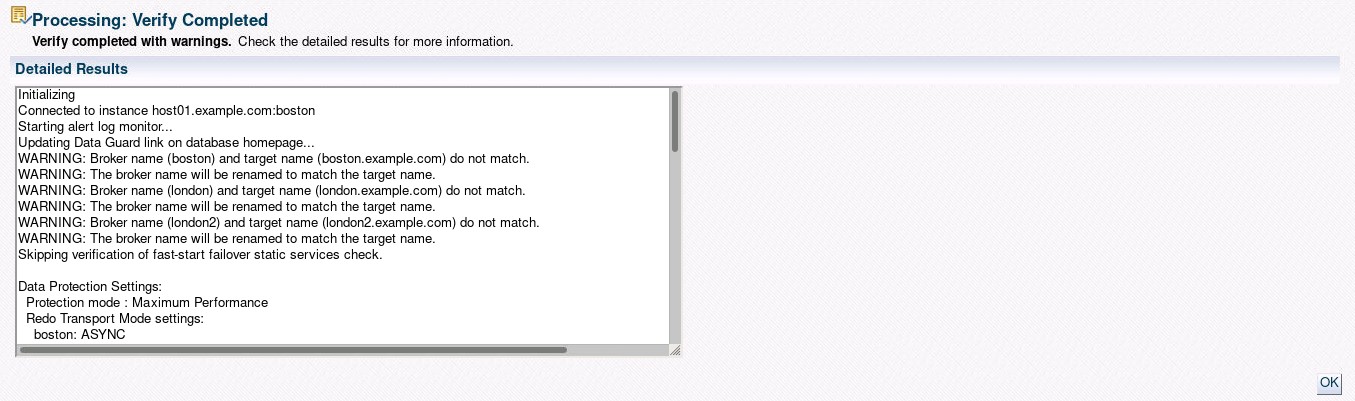


1. Select the **Verify Configuration** item from the menu. The following image shows the steps performed while verifying the configuration. After verification completes, detailed results are displayed.



**Note:** The verify process will complete with warnings. You can safely ignore these warnings at this time.

1. Click **OK**. The Data Guard Administration page is displayed.



# Practices for Lesson 11: Monitoring a Data Guard Broker Configuration

## Practices for Lesson 11: Overview

### Practices Overview

In these practices, you will use the DGMGRL utility to monitor your physical standby database. You will also examine the use of trace files to monitor the Data Guard environment.

## Practice 11-1: Monitoring the Physical Standby Database

### Overview

In this practice, you will use DGMGRL and connect with both local and remote connections. The password file will be updated on the primary database and copied to every other destination in the Data Guard configuration.

### Tasks

1. Use a terminal window on localhost connected as oracle with the environment variables set to orclcdb. Launch DGMGRL connecting as the SYSDG user with operating system authentication.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@localhost ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Thu Jun 4 18:52:53 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg/<password>@orclcdb** Connected to "orclcdb"

Connected as SYSDG.

DGMGRL>

1. Use the SHOW CONFIGURATION VERBOSE command to display the current values for the

CommunicationTimeout property and the OperationTimeout property.

DGMGRL> **show configuration verbose**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby - Physical standby database stndby2 - Logical standby database

|  |  |  |
| --- | --- | --- |
| FastStartFailoverThreshold | = | '30' |
| **OperationTimeout** | **=** | **'30'** |
| TraceLevel | = | 'USER' |
| FastStartFailoverLagLimit | = | '30' |
| **CommunicationTimeout** | **=** | **'180'** |
| ObserverReconnect | = | '0' |
| FastStartFailoverAutoReinstate | = | 'TRUE' |
| FastStartFailoverPmyShutdown | = | 'TRUE' |
| BystandersFollowRoleChange | = | 'ALL' |
| ObserverOverride | = | 'FALSE' |
| ExternalDestination1 | = | '' |
| ExternalDestination2 | = | '' |
| PrimaryLostWriteAction | = | 'CONTINUE' |
| ConfigurationWideServiceName | = | 'orclcdb\_CFG' |

1. Modify the CommunicationTimeout property and set it to a value of 300. Verify the 300result.

DGMGRL> **edit configuration set property 'CommunicationTimeout' = 300;**

Property "CommunicationTimeout" updated

DGMGRL> **show configuration verbose**

...

Properties:

FastStartFailoverThreshold = '30'

OperationTimeout = '30'

TraceLevel = 'USER'

FastStartFailoverLagLimit = '30'

**CommunicationTimeout = '300'**

ObserverReconnect = '0'

Members Not Receiving Redo: stndbyFS - Far sync instance

Properties:

Fast-Start Failover: Disabled

Configuration Status: SUCCESS

DGMGRL>

FastStartFailoverAutoReinstate = 'TRUE'

...

**Note:** This is not normally needed but it helps with labs running in the Virtual Machine architecture.

1. Modify the OperationTimeout property and set it to the maximum value of 300. Verify the result.

DGMGRL> **edit configuration set property 'OperationTimeout' = 300;**

Property "CommunicationTimeout" updated

DGMGRL> **show configuration verbose**

...

Properties:

FastStartFailoverThreshold = '30'

**OperationTimeout = '300'**

TraceLevel = 'USER'

FastStartFailoverLagLimit = '30'

CommunicationTimeout = '300'

ObserverReconnect = '0' FastStartFailoverAutoReinstate = 'TRUE'

...

1. Use the SHOW DATABASE command for the physical standby database and determine the current transport lag, apply lag, and apply rate.

DGMGRL> **show database stndby**

Database - stndby

Role:

Intended State: Transport Lag:

Apply Lag:

PHYSICAL STANDBY APPLY-ON

0 seconds (computed 0 seconds ago)

0 seconds (computed 0 seconds ago)

Average Apply Rate: 17.00 KByte/s Real Time Query: OFF Instance(s):

stndby

Database Status: SUCCESS

DGMGRL>

1. Stop redo apply on the physical standby database to force an apply rate lag to occur.

DGMGRL> **edit database stndby set state = 'APPLY-OFF';**

Succeeded.

1. Without exiting DGMGRL, force a log switch on the primary database.

**Note:** You are currently connected to the primary database.

DGMGRL> **SQL "alter system switch logfile";**

Succeeded.

1. Use the SHOW DATABASE command for the physical standby database and display the current apply lag rate.

DGMGRL> **show database stndby**

Database - stndby

Role: PHYSICAL STANDBY

Intended State: APPLY-OFF

Transport Lag: 0 seconds (computed 0 seconds ago) Apply Lag: 33 seconds (computed 0 seconds ago) Apply Rate: (unknown)

Real Time Query: OFF Instance(s):

stndby

Database Status:

SUCCESS

1. Display the standby receive queue for the physical standby database.

DGMGRL> **show database stndby 'RecvQEntries';**

STANDBY\_RECEIVE\_QUEUE

STATUS RESETLOGS\_ID

THREAD

LOG\_SEQ FIRST\_CHANGE#

TIME\_GENERATED

TIME\_COMPLETED

NEXT\_CHANGE#

1042035828

SIZE (KBs)

PARTIALLY\_APPLIED

1

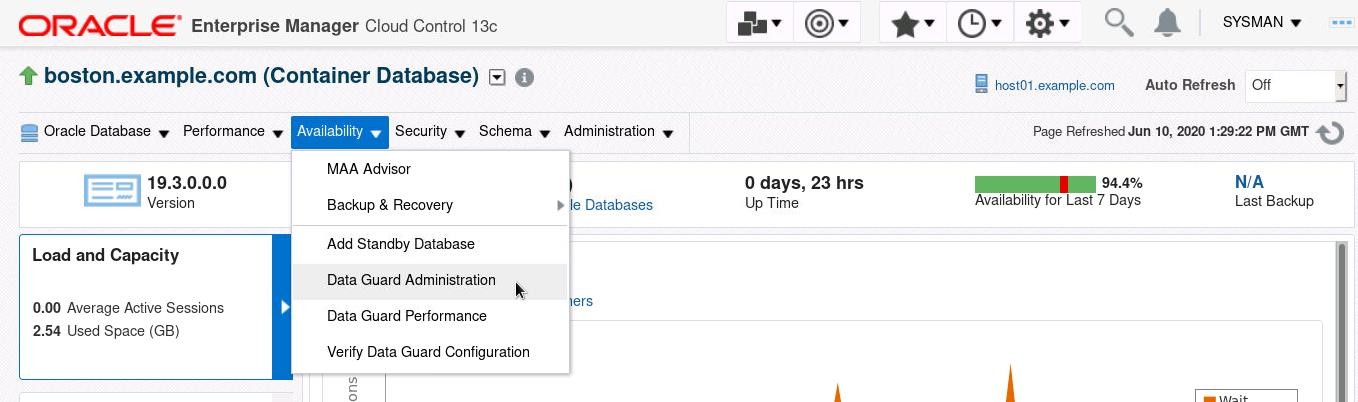
63 06/04/2020 18:55:17 06/04/2020 19:03:47

3030103 3032604 4151

DGMGRL>

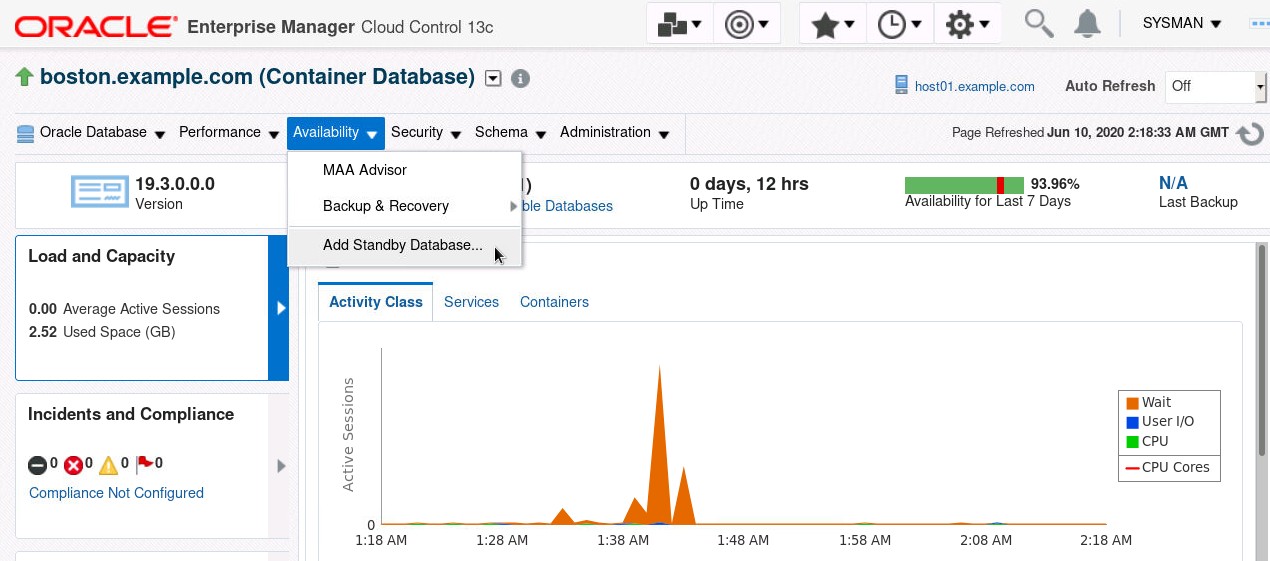
1. (**Optional**) With the navigation techniques learned in practice 10-3, navigate to the orclcdb database home page. On the orclcdb database home page, click **Data Guard Administration** in the Availability menu.

orclcdb



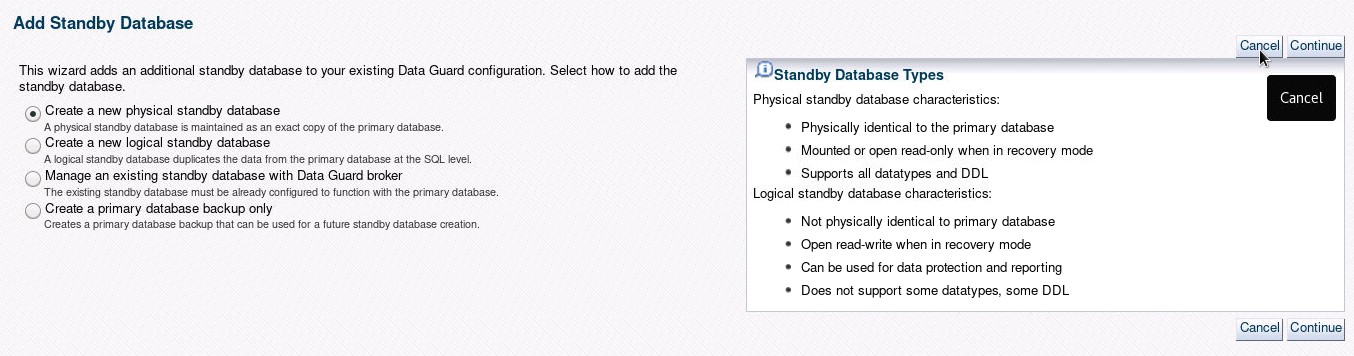
**(Optional)** If only the Add Standby Database link is visible, then select it. It will not launch the Add Standby Database Wizard, but instead will navigate to the Data Guard home page.

orclcdb

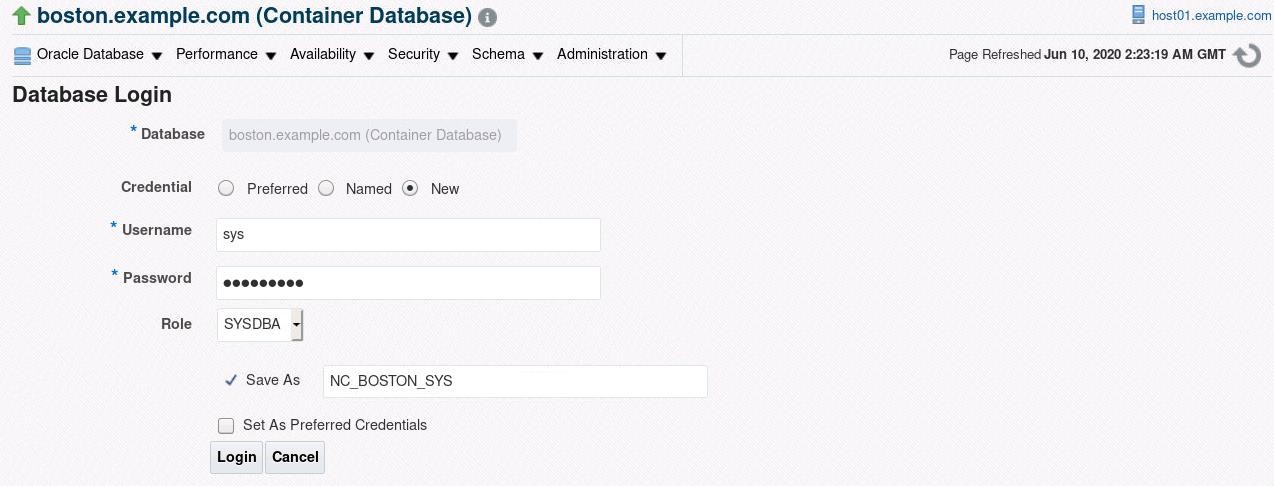


**(Optional)** If the Add Standby Database link shows the Add Standby Database page, click

**Cancel** to navigate to the Data Guard home page.



1. (**Optional**) On the Database Login Page, select **New** in the Credential option with the following values. Click **Login**.
   * Username: sys
   * Password: *<password>*
   * Role: SYSDBA
   * Save As: NC\_ORCLCDB\_SYS



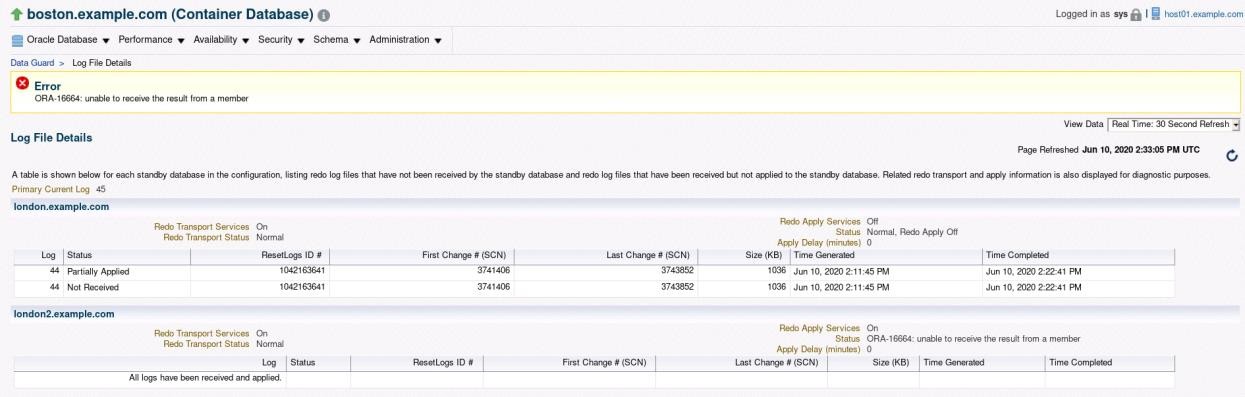
1. (**Optional**) On the Data Guard home page, you will see the status of redo apply service on the physical standby database



1. (**Optional**) Click the Log File Details link in the Performance section.
2. (**Optional**) On the Log File Details page, you will see the list of log files that have not been received and applied.

**Note:** The ORA-16664 message can be ignored.

orclcdb



1. Restart redo apply on the physical standby database. Verify that the apply lag has been cleared. Exit DGMGRL when done.

**Note:** You may have to wait a minute after restarting redo apply to verify the results.

DGMGRL> **edit database stndby set state = 'APPLY-ON';**

Succeeded.

DGMGRL> **show database stndby**

Database - stndby

Role: PHYSICAL STANDBY

Intended State: APPLY-ON

Transport Lag: 0 seconds (computed 0 seconds ago) Apply Lag: 0 seconds (computed 0 seconds ago) Apply Rate: 0 Byte/s

Real Time Query: OFF Instance(s):

stndby

Database Status:

SUCCESS

DGMGRL> **exit;**

## Practice 11-2: Examining Data Guard Log and Trace Files

### Overview

In this practice, you will locate and examine the Data Guard log and trace files.

### Tasks

1. Use a terminal window on localhost connected as oracle with the environment variables set to orclcdb. Connect to the primary database using SQL\*Plus and determine the root directory for the Automatic Diagnostic Repository (ADR). Exit SQL\*Plus when done.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Thu Jun 4 19:06:34 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **show parameter diag**

NAME

TYPE

VALUE

diagnostic\_dest

string

/u01/app/oracle

SQL> **exit;**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@localhost ~]$

1. Change directory to the "trace" subdirectory located in the Automatic Diagnostic Repository home location. The ADR home is located at

<diagnostic\_dest>/diag/rdbms/<dbname>/<instance\_name>.

[oracle@localhost]$ **cd**

**/u01/app/oracle/diag/rdbms/orclcdb/orclcdb/trace**

1. Verify that the previous commands that changed the state of redo apply and connection timeout where recorded in the Data Guard broker log file. The broker log file is named drc<db\_unique\_name>.log.

[oracle@localhost trace]$ **grep CommunicationTimeout drcorclcdb.log** EDIT CONFIGURATION SET PROPERTY CommunicationTimeout = 300 Property "CommunicationTimeout" value set to "300" seconds

EDIT CONFIGURATION SET PROPERTY CommunicationTimeout = 300 completed successfully

[oracle@localhost trace]$ **grep APPLY drcorclcdb.log**

EDIT DATABASE stndby SET STATE = APPLY-OFF

EDIT DATABASE stndby SET STATE = APPLY-OFF completed successfully

EDIT DATABASE stndby SET STATE = APPLY-ON

EDIT DATABASE stndby SET STATE = APPLY-ON completed successfully

[oracle@localhost trace]$

1. Use the "ls -alt | more" command to list the directory contents of the trace directory sorted by modification time descending. The most recent modified file will be displayed first. Use <Ctrl + C> to exit.

[oracle@localhost trace]$ **ls -alt | more**

total 17812

-rw-r-----. 1 oracle oinstall 3130 Jun 4 19:09 orclcdb\_m000\_7209.trc

-rw-r-----. 1 oracle oinstall 1015 Jun 4 19:09 orclcdb\_m000\_7209.trm

-rw-r-----. 1 oracle oinstall 12215 Jun 4 19:09 orclcdb\_m001\_5486.trc

-rw-r-----. 1 oracle oinstall 1602 Jun 4 19:09 orclcdb\_m001\_5486.trm

-rw-r-----. 1 oracle oinstall 14706 Jun 4 19:09 orclcdb\_mmon\_5423.trc

-rw-r-----. 1 oracle oinstall 1594 Jun 4 19:09 orclcdb\_mmon\_5423.trm

-rw-r-----. 1 oracle oinstall 11207 Jun 4 19:08 orclcdb\_m004\_5819.trc

...

<Ctrl + C>

[oracle@localhost trace]$

1. Connect to the primary database using SQL\*Plus and set the level to 16 to track detailed archived redo log destination activity. Force a log switch and exit SQL\*Plus when done.

[oracle@localhost trace]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Thu Jun 4 19:10:08 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter system set log\_archive\_trace=16;**

System altered.

SQL> **alter system switch logfile;**

System altered.

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@localhost trace]$

1. Use the "ls -alt | more" command to list the directory contents of the trace directory sorted by modification time descending. Identify the newly created files in the directory that were not present for the previous step 4. Use <Ctrl + C> to exit.

[oracle@localhost trace]$ **ls -alt | more**

total 18152

-rw-r-----. 1 oracle oinstall 181546 Jun 4 19:11 orclcdb\_lgwr\_5343.trc

-rw-r-----. 1 oracle oinstall 2818 Jun 4 19:11 orclcdb\_lgwr\_5343.trm

-rw-r-----. 1 oracle oinstall 165194 Jun 4 19:11 orclcdb\_nss2\_5502.trc

-rw-r-----. 1 oracle oinstall 2675 Jun 4 19:11 orclcdb\_nss2\_5502.trm

...

<Ctrl + C> [oracle@localhost trace]$

1. The resulting log writer process (LGWR) and network server sync process (NSS) trace files can be very large in size. Since the primary database is configured for SYNC redo transport to the orclcdbFS far sync destination, verify that communication occurred to this destination in the trace file using the "grep orclcdbFS <NSS2 trace file name>" command. The trace file name must be determined from the previous step. You may explore the contents of the trace files if desired.

[oracle@localhost trace]$ **grep orclcdb orclcdb\_nss2\_5343.trc**

\*rfsnam: /u01/app/oracle/oradata/orclcdb/stdbyredo02.log [oracle@localhost trace]$

1. Connect to the primary database using SQL\*Plus and set the log\_archive\_trace level to 0 to disable tracing. Exit SQL\*Plus when done.

[oracle@localhost trace]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Thu Jun 4 19:13:55 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter system set log\_archive\_trace=0;**

System altered.

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@localhost trace]$

## Practice 11-3: Using the VALIDATE commands

### Overview

In this practice, you will look at the various ways to use the DGMGRL VALIDATE commands.

### Tasks

1. Use the terminal session connected to localhost as the oracle user and set the environment.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [orclcdb] ? **orclcdb**

The Oracle base remains unchanged with value /u01/app/oracle [oracle@localhost ~]$

1. Using DGMGRL, connect to the orclcdb database.

[oracle@localhost ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Thu Jun 4 21:45:46 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg/<password>@orclcdb** Connected to "orclcdb"

Connected as SYSDG. DGMGRL>

1. Use the help command to display all possible VALIDATE commands.

DGMGRL> **help VALIDATE**

Performs an exhaustive set of validations for a member Syntax:

VALIDATE DATABASE [VERBOSE] <database name>;

VALIDATE DATABASE [VERBOSE] <database name> DATAFILE <datafile number>

OUTPUT=<file name>;

VALIDATE DATABASE [VERBOSE] <database name> SPFILE;

VALIDATE FAR\_SYNC [VERBOSE] <far\_sync name> [WHEN PRIMARY IS <database name>];

VALIDATE NETWORK CONFIGURATION FOR { ALL | <member name> };

VALIDATE STATIC CONNECT IDENTIFIER FOR { ALL | <database

name> };

DGMGRL>

1. Use the VALIDATE command to perform a comparison of SPFILE entries between the

orclcdb primary database and the stndby standby database.

DGMGRL> **VALIDATE DATABASE stndby SPFILE;**

Connecting to "orclcdb". Connected to "orclcdb"

Connecting to "stndby". Connected to "stndby"

**Parameter settings with different values:**

audit\_file\_dest:

orclcdb (PRIMARY) : /u01/app/oracle/admin/orclcdb/adump stndby : /u01/app/oracle/admin/stndby/adump

dispatchers:

orclcdb (PRIMARY) : (PROTOCOL=TCP) (SERVICE=orclcdbXDB) stndby : (PROTOCOL=TCP) (SERVICE=stndbyXDB)

enabled\_PDBs\_on\_standby:

orclcdb (PRIMARY) : NOT SPECIFIED

stndby : \*

log\_archive\_trace: orclcdb (PRIMARY) : 0 stndby

: NOT SPECIFIED

DGMGRL>

**Note:** The command above shows only the parameter settings with different values. If you

want to list all parameter settings compared, use VALIDATE DATABASE VERBOSE stndby SPFILE.

1. Validate network configuration for the stndby database.

DGMGRL> **VALIDATE NETWORK CONFIGURATION FOR stndby;**

Connecting to instance "stndby" on database "stndby" ... Connected to "stndby"

Checking connectivity from instance "stndby" on database "stndby to instance "orclcdb" on database "orclcdb"...

Succeeded.

Checking connectivity from instance "stndby" on database "stndby to instance "orclcdbFS" on database "orclcdbFS"...

Succeeded.

Checking connectivity from instance "stndby" on database "stndby to instance "stndby2" on database "stndby2"...

Succeeded.

Checking connectivity from instance "stndby" on database "stndby to instance "stndbyFS" on database "stndbyFS"...

Succeeded.

Connecting to instance "orclcdb" on database "orclcdb" ... Connected to "orclcdb"

Checking connectivity from instance "orclcdb" on database "orclcdb to instance "stndby" on database "stndby"...

Succeeded.

Connecting to instance "orclcdbFS" on database "orclcdbFS" ... Connected to "orclcdbFS"

Checking connectivity from instance "orclcdbFS" on database "orclcdbFS to instance "stndby" on database "stndby"...

Succeeded.

Connecting to instance "stndby2" on database "stndby2" ... Connected to "stndby2"

Checking connectivity from instance "stndby2" on database "stndby2 to instance "stndby" on database "stndby"...

Succeeded.

Connecting to instance "stndbyFS" on database "stndbyFS" ... Connected to "stndbyFS"

Checking connectivity from instance "stndbyFS" on database "stndbyFS to instance "stndby" on database "stndby"...

Succeeded.

Oracle Clusterware is not configured on database "stndby". Connecting to database "stndby" using static connect identifier "(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=stndby.example.com)(POR T=1521))(CONNECT\_DATA=(SERVICE\_NAME=stndby\_DGMGRL.example.com)(INS TANCE\_NAME=stndby)(SERVER=DEDICATED)(STATIC\_SERVICE=TRUE)))" ...

Succeeded.

The static connect identifier allows for a connection to database "stndby".

DGMGRL>

1. Validate network configuration for all members.

#### DGMGRL> VALIDATE NETWORK CONFIGURATION FOR all;

Connecting to instance "orclcdb" on database "orclcdb" ... Connected to "orclcdb"

Checking connectivity from instance "orclcdb" on database "orclcdb to instance "orclcdbFS" on database "orclcdbFS"...

Succeeded.

Checking connectivity from instance "orclcdb" on database "orclcdb to instance "stndby" on database "stndby"...

Succeeded.

Checking connectivity from instance "orclcdb" on database "orclcdb to instance "stndby2" on database "stndby2"...

Succeeded.

Checking connectivity from instance "orclcdb" on database "orclcdb to instance "stndbyFS" on database "stndbyFS"...

Succeeded.

Connecting to instance "orclcdbFS" on database "orclcdbFS" ... Connected to "orclcdbFS"

Checking connectivity from instance "orclcdbFS" on database "orclcdbFS to instance "orclcdb" on database "orclcdb"...

Succeeded.

Checking connectivity from instance "orclcdbFS" on database "orclcdbFS to instance "stndby" on database "stndby"...

Succeeded.

Checking connectivity from instance "orclcdbFS" on database "orclcdbFS to instance "stndby2" on database "stndby2"...

Succeeded.

Checking connectivity from instance "orclcdbFS" on database "orclcdbFS to instance "stndbyFS" on database "stndbyFS"... Succeeded.

Connecting to instance "stndby" on database "stndby" ... Connected to "stndby"

Checking connectivity from instance "stndby" on database "stndby to instance "orclcdb" on database "orclcdb"...

Succeeded.

Checking connectivity from instance "stndby" on database "stndby to instance "orclcdbFS" on database "orclcdbFS"...

Succeeded.

Checking connectivity from instance "stndby" on database "stndby to instance "stndby2" on database "stndby2"...

Succeeded.

Checking connectivity from instance "stndby" on database "stndby to instance "stndbyFS" on database "stndbyFS"...

Succeeded.

Connecting to instance "stndby2" on database "stndby2" ... Connected to "stndby2"

Checking connectivity from instance "stndby2" on database "stndby2 to instance "orclcdb" on database "orclcdb"...

Succeeded.

Checking connectivity from instance "stndby2" on database "stndby2 to instance "orclcdbFS" on database "orclcdbFS"...

Succeeded.

Checking connectivity from instance "stndby2" on database "stndby2 to instance "stndby" on database "stndby"...

Succeeded.

Checking connectivity from instance "stndby2" on database "stndby2 to instance "stndbyFS" on database "stndbyFS"...

Succeeded.

Connecting to instance "stndbyFS" on database "stndbyFS" ... Connected to "stndbyFS"

Checking connectivity from instance "stndbyFS" on database "stndbyFS to instance "orclcdb" on database "orclcdb"...

Succeeded.

Checking connectivity from instance "stndbyFS" on database "stndbyFS to instance "orclcdbFS" on database "orclcdbFS"... Succeeded.

Checking connectivity from instance "stndbyFS" on database "stndbyFS to instance "stndby" on database "stndby"...

Succeeded.

Checking connectivity from instance "stndbyFS" on database "stndbyFS to instance "stndby2" on database "stndby2"...

Succeeded.

Oracle Clusterware is not configured on database "orclcdb". Connecting to database "orclcdb" using static connect identifier "(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=localhost.example.com)(POR T=1521))(CONNECT\_DATA=(SERVICE\_NAME=orclcdb\_DGMGRL.example.com)(INS TANCE\_NAME=orclcdb)(SERVER=DEDICATED)(STATIC\_SERVICE=TRUE)))" ...

Succeeded.

The static connect identifier allows for a connection to database "orclcdb".

Oracle Clusterware is not configured on database "stndby". Connecting to database "stndby" using static connect identifier "(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=stndby.example.com)(POR T=1521))(CONNECT\_DATA=(SERVICE\_NAME=stndby\_DGMGRL.example.com)(INS TANCE\_NAME=stndby)(SERVER=DEDICATED)(STATIC\_SERVICE=TRUE)))" ...

Succeeded.

The static connect identifier allows for a connection to database "stndby".

Oracle Clusterware is not configured on database "stndby2". Connecting to database "stndby2" using static connect identifier "(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=stndby.example.com)(POR T=1521))(CONNECT\_DATA=(SERVICE\_NAME=stndby2\_DGMGRL.example.com)(IN STANCE\_NAME=stndby2)(SERVER=DEDICATED)(STATIC\_SERVICE=TRUE)))" ...

Succeeded.

The static connect identifier allows for a connection to database "stndby2".

DGMGRL>

1. Validate the static connect identifier of the orclcdb database.

DGMGRL> **VALIDATE STATIC CONNECT IDENTIFIER FOR orclcdb;**

Oracle Clusterware is not configured on database "orclcdb". Connecting to database "orclcdb" using static connect identifier "(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=localhost.example.com)(POR T=1521))(CONNECT\_DATA=(SERVICE\_NAME=orclcdb\_DGMGRL.example.com)(INS TANCE\_NAME=orclcdb)(SERVER=DEDICATED)(STATIC\_SERVICE=TRUE)))" ...

Succeeded.

The static connect identifier allows for a connection to database "orclcdb".

DGMGRL>

1. Validate the static connect identifier for all databases.

DGMGRL> **VALIDATE STATIC CONNECT IDENTIFIER FOR all;**

Oracle Clusterware is not configured on database "orclcdb". Connecting to database "orclcdb" using static connect identifier "(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=localhost.example.com)(POR T=1521))(CONNECT\_DATA=(SERVICE\_NAME=orclcdb\_DGMGRL.example.com)(INS TANCE\_NAME=orclcdb)(SERVER=DEDICATED)(STATIC\_SERVICE=TRUE)))" ...

Succeeded.

The static connect identifier allows for a connection to database "orclcdb".

Oracle Clusterware is not configured on database "stndby". Connecting to database "stndby" using static connect identifier "(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=stndby.example.com)(POR

T=1521))(CONNECT\_DATA=(SERVICE\_NAME=stndby\_DGMGRL.example.com)(INS TANCE\_NAME=stndby)(SERVER=DEDICATED)(STATIC\_SERVICE=TRUE)))" ...

Succeeded.

The static connect identifier allows for a connection to database "stndby".

Oracle Clusterware is not configured on database "stndby2". Connecting to database "stndby2" using static connect identifier "(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=stndby.example.com)(POR T=1521))(CONNECT\_DATA=(SERVICE\_NAME=stndby2\_DGMGRL.example.com)(IN STANCE\_NAME=stndby2)(SERVER=DEDICATED)(STATIC\_SERVICE=TRUE)))" ...

Succeeded.

The static connect identifier allows for a connection to database "stndby2".

DGMGRL>

1. Exit DGMGRL on localhost leaving the terminal window open for future practices.

# Practices for Lesson 12: Configuring Data Protection Modes

## Practices for Lesson 12: Overview

### Practices Overview

In these practices, you will examine the various protection modes and the impact that they may have on the primary database.

## Practice 12-1: Examining the Maximum Availability Protection Mode

### Overview

In this practice, you will use DGMGRL to view the current protection mode and modify it to maximum availability. You will simulate a problem on the standby database and observe the impact if any to the primary database.

### Tasks

1. Use a terminal window on localhost connected as oracle with the environment variables set to orclcdb. Launch the DGMGRL utility and connect as the sysdg user with operating system authentication.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@localhost]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Tue Jun 16 02:37:54 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg/<password>@orclcdb** Connected to "orclcdb"

Connected as SYSDG.

1. Use the SHOW CONFIGURATION command to display the current protection mode for the Data Guard configuration.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby - Physical standby database stndby2 - Logical standby database

Members Not Receiving Redo:

stndbyFS - Far sync instance Fast-Start Failover: Disabled

Configuration Status:

SUCCESS (status updated 58 seconds ago)

DGMGRL>

1. Using DGMGRL, determine the current LogXptMode for the far sync instance and the physical standby database. EXAMPLE for FAR SYNC applications. DO NOT EXECUTE

DGMGRL> **show database 'orclcdb' 'LogXptMode';**

LogXptMode = 'ASYNC';

DGMGRL> **show database stndby 'LogXptMode';**

LogXptMode = 'ASYNC'; DGMGRL> **exit**

[oracle@localhost ~]$

**Note:** Database names, far sync instance names, property names and property values do not need to be enclosed in single quotation marks unless they are case-sensitive. The documentation examples generally use a single quote for both property names and property values and that convention has been followed in these labs. For the above example, only the far sync name 'orclcdbFS' would require it to be enclosed by single quotes. The second edit command does not use any single quotes for illustration.

1. ***Optional: DO NOT EXECUTE this example:*** If the current LogXptMode is not reported as 'ASYNC' for both the far sync instance and the physical standby database, then explicitly set it to 'ASYNC'. Exit DGMGRL when done.

DGMGRL> **edit far\_sync 'orclcdbFS' set property 'LogXptMode' = 'ASYNC';**

Property "LogXptMode" updated

DGMGRL> **edit database stndby set property LogXptMode = ASYNC;**

Property "logxptmode" updated

DGMGRL> **exit**

[oracle@localhost ~]$

1. Connect to the primary database using SQL\*Plus and determine the current value for the

LOG\_ARCHIVE\_DEST\_2 parameter. What is the current LogXptMode? Exit SQL\*Plus.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 10:57:04 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **show parameter LOG\_ARCHIVE\_DEST\_2**

NAME

TYPE

VALUE

log\_archive\_dest\_2 string

service="orclcdbfs", ***SYNC AFFIRM*** delay=0 optional compression=disable max\_failure=0 max\_connections=1 reopen=300 db\_unique\_name="orclcdbFS" net\_timeout=30, valid\_for=

(online\_logfile, all\_roles)

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@localhost ~]$

1. Launch the DGMGRL utility and connect as the SYSDG user.

[oracle@localhost ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Fri Jun 5 10:58:40 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg/<password>@orclcdb** Connected to "orclcdb"

Connected as SYSDG.

DGMGRL>

1. Display the value for the RedoRoutes property of the primary database.

DGMGRL> **show database 'orclcdb' 'RedoRoutes';**

RedoRoutes = ''

**Note:** When the property 'RedoRoutes' has been defined, it takes precedence over the value of the property 'LogXptMode'. The property 'LogXptMode' will continue to report 'ASYNC' even though the actual transport mode is currently 'SYNC'.

1. Modify the 'RedoRoutes' property for the orclcdb primary database and set it to the 'ASYNC' redo transport mode.

DGMGRL> **edit database orclcdb set property 'RedoRoutes' = '(orclcdb:stndby ASYNC)';**

Property "RedoRoutes" updated

1. Attempt to change the configuration mode to maximum availability and notice the results.

DGMGRL> **edit configuration set protection mode as maxavailability;**

Error: ORA-16627: operation disallowed since no member would remain to support protection mode

Failed.

1. Modify the RedoRoutes property for the orclcdb primary database and set it to the 'FASTSYNC' redo transport mode.

DGMGRL> **edit database orclcdb set property 'RedoRoutes' = '(orclcdb:stndby FASTSYNC)';**

Property "RedoRoutes" updated

1. Change the configuration mode to maximum availability and verify the results.

DGMGRL> **edit configuration set protection mode as maxavailability;**

Succeeded.

DGMGRL> **show configuration;**

Configuration - DRSolution

Protection Mode: MaxAvailability Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby - Physical standby database stndby2 - Logical standby database

Members Not Receiving Redo: stndbyFS - Far sync instance

Fast-Start Failover: Disabled Configuration Status:

SUCCESS (status updated 60 seconds ago)

DGMGRL>

1. Use a terminal window on stndby connected as oracle with the environment variables set to stndby. Connect to the physical standby database using SQL\*Plus and perform a shutdown abort.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base remains unchanged with value /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 11:04:00 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **shutdown abort**

ORACLE instance shut down.

SQL>

1. Return to the DGMGRL session running on localhost and display the configuration.

DGMGRL> **show configuration**

* The messages will be similar not exact.sho

Configuration - DRSolution

Protection Mode: MaxAvailability Members:

orclcdb - Primary database orclcdbFS - Far sync instance

Error: ORA-16778: redo transport error for one or more members

stndby - Physical standby database Error: ORA-1034: ORACLE not available

stndby2 - Logical standby database

Members Not Receiving Redo: stndbyFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:

ERROR (status updated 55 seconds ago)

DGMGRL>

1. Return to the SQL\*Plus session on stndby connected as oracle with the environment variables set to stndby. Use SQL\*Plus to restart and mount the physical standby database. Verify that the DEV1 pluggable database is also mounted.

SQL> **startup mount**

ORACLE instance started.

Total System Global Area 517763072 bytes Fixed Size 2290216 bytes Variable Size 440405464 bytes

Database Buffers 71303168 bytes

Redo Buffers 3764224 bytes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Database mounted.  SQL> **show pdbs** | | | | | | |
| CON\_ID |  | CON\_NAME |  | OPEN MODE |  | RESTRICTED |
| 2 |  | PDB$SEED |  | MOUNTED |  |  |
| 3 |  | DEV1 |  | MOUNTED |  |  |

1. Return to the DGMGRL session running on localhost and display the configuration.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxAvailability Members:

orclcdb - Primary database

Error: ORA-16778: redo transport error for one or more members

stndby - Physical standby database

Warning: ORA-16809: multiple warnings detected for the

member

stndby2 - Logical standby database Members Not Receiving Redo:

stndbyFS - Far sync instance Fast-Start Failover: Disabled

Configuration Status:

ERROR (status updated 17 seconds ago)

**Note:** The broker may have restarted redo apply before you are able to see the above error. In addition, you may also receive a series of warnings ORA-16778 and ORA-16809 standby disconnected from redo source for longer than specified threshold. This is acceptable.

1. Restart redo apply for the physical standby database. Perform a log switch on the primary database and verify the configuration.

DGMGRL> **edit database stndby set state = 'APPLY-ON';**

Succeeded.

DGMGRL> **SQL "alter system switch logfile";**

Succeeded.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxAvailability Members:

orclcdb - Primary database selexcshowinstance

stndby - Physical standby database stndby2 - Logical standby database

Fast-Start Failover: Disabled

Configuration Status:

SUCCESS (status updated 56 seconds ago)

DGMGRL>

1. Before proceeding with additional lab steps, give the transport lag and apply lag an opportunity to catch up. Use the show configuration and show database stndby commands until the lag clears. Repeat these commands as needed.

DGMGRL> **show database stndby**

Database - stndby

Role: PHYSICAL STANDBY

Intended State: APPLY-ON

Transport Lag: **0 seconds** (computed 1 second ago) Apply Lag: **0 seconds** (computed 1 second ago) Apply Rate: 0 Byte/s

Real Time Query: OFF Instance(s):

stndby

## Practice 12-2: Examining the Maximum Protection Mode

### Overview

In this practice, you will use DGMGRL to modify the current protection mode to maximum protection. You will simulate a problem on the standby database and observe the impact to the primary database.

### Tasks

1. Modify the RedoRoutes property for the orclcdb primary database and set it to the SYNC redo transport mode. Enable the maximum protection mode for the Data Guard configuration.

DGMGRL> **edit database orclcdb set property 'RedoRoutes' = '(orclcdb:stndby SYNC)';**

Property "RedoRoutes" updated

DGMGRL> **edit configuration set protection mode as maxprotection;**

Error: ORA-16627: operation disallowed since no standby databases would remain to support protection mode

NOTE: If you have 2 standby databases up it will succeed.

Failed.

**Note:** The maximum protection mode is not supported by far sync.

1. devshow

DGMGRL> **edit configuration set protection mode as maxprotection;**

Succeeded.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxProtection Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby2 - Logical standby database stndby - Physical standby database

Members Not Receiving Redo: stndbyFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:

SUCCESS (status updated 57 seconds ago)

DGMGRL> **exit**

[oracle@localhost ~]$

1. staEnable the maximum protection mode for the Data Guard configuration and display the resulting configuration. Exit DGMGRL.
2. In the same terminal window on localhost, connect to the primary database using SQL\*Plus and switch the session to the DEV1 pluggable database. Leave this window open.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 11:19:44 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter session set container=DEV1;**

Session altered.

SQL>

1. Use a terminal window on stndby connected as oracle with the environment variables set to stndby. Connect to the physical standby database using SQL\*Plus and perform a shutdown abort.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base has been set to /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 11:21:26 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **shutdown abort**

ORACLE instance shut down. SQL>

1. Return to the SQL\*Plus session on localhost with the session set to the DEV1 pluggable database. Display the current data for the HR.REGIONS table and then insert a new row into the table. Exit the terminated session.

SQL> **select \* from hr.regions order by region\_id;**

REGION\_ID REGION\_NAME

1. Europe
2. Americas
3. Asia
4. Middle East and Africa
5. Australia

SQL> **insert into hr.regions values (6,'MyRegion');**

insert into hr.regions values (6,'MyRegion')

\*

ERROR at line 1:

ORA-03135: connection lost contact Process ID: 19624

Session ID: 20 Serial number: 3265

**Note:** If the row inserts successfully, then attempt to commit the change. After waiting for about 5 minutes, you will receive ORA-03113 error message.

1 row created.

SQL> **commit;**

ERROR at line 1:

ORA-03113: end-of-file on communication channel Process ID: 15203

Session ID: 65 Serial number: 2297

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@localhost ~]$

**Note:** The primary database has been brought down due to the maximum protection mode and not having the standby database available to accept redo. Depending on timings and blocks cached in memory, the insert may be successful, but the commit will always fail. You may have to wait for the timeout period to elapse before seeing the error message.

1. In the same terminal window on localhost, connect to the primary database using SQL\*Plus and attempt to restart the instance. Exit the terminated session.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 11:30:15 2020

Version 19.3.0.0.0

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

Connected to an idle instance.

SQL> **startup**

ORACLE instance started.

Total System Global Area 629145352 bytes

|  |  |  |
| --- | --- | --- |
| Fixed Size | 9137928 | bytes |
| Variable Size | 373293056 | bytes |
| Database Buffers | 239075328 | bytes |
| Redo Buffers | 7639040 | bytes |
| Database mounted. |  |  |

ORA-03113: end-of-file on communication channel Process ID: 11692

Session ID: 19 Serial number: 51864

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0 [oracle@localhost ~]$

**Note:** The primary cannot be started with the physical standby down with the maximum protection mode and no other standby databases available that support this mode.

1. Return to the SQL\*Plus session on stndby to start up and mount the physical standby database. Verify that the DEV1 pluggable database is mounted. Exit SQL\*Plus.

SQL> **startup mount**

ORACLE instance started.

Total System Global Area 629145352 bytes

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@stndby ~]$

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fixed Size | 9137928 | bytes |  | |
| Variable Size | 369098752 | bytes |
| Database Buffers | 243269632 | bytes |
| Redo Buffers | 7639040 | bytes |
| Database mounted.  SQL> **show pdbs** |  |  |
| CON\_ID CON\_NAME |  | OPEN MODE |  | RESTRICTED |
| 2 PDB$SEED |  | MOUNTED |  |  |
| 3 DEV1 |  | MOUNTED |  |  |

1. Return to the terminal window on localhost connected as oracle with the environment variables set to orclcdb. Use SQL\*Plus to open the primary database. Verify that the pluggable database is open. If not, then open it. Exit SQL\*Plus.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 11:34:12 2020

Version 19.3.0.0.0

(c) 1982, 2019, Oracle. All rights reserved. Connected to an idle instance.

SQL> **startup**

ORACLE instance started.

Total System Global Area 629145352 bytes

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@localhost ~]$

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fixed Size | 9137928 | bytes |  | | |
| Variable Size | 373293056 | bytes |
| Database Buffers | 239075328 | bytes |
| Redo Buffers Database mounted. Database opened. | 7639040 | bytes |
| SQL> **show pdbs** |  |  |
| CON\_ID CON\_NAME |  | OPEN | MODE |  | RESTRICTED |
| 2 PDB$SEED |  | READ | ONLY |  | NO |
| 3 DEV1 |  | READ | WRITE |  |  |

1. In the same terminal window on localhost, launch the DGMGRL utility and connect as the SYSDG user. Display the Data Guard configuration. Wait until the ORA-\* warning messages are cleared.

[oracle@localhost ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Fri Jun 5 11:37:13 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg/<password>@orclcdb** Connected to "orclcdb"

Connected as SYSDG. DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxProtection Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby2 - Logical standby database stndby - Physical standby database

Members Not Receiving Redo:

Fast-Start Failover: Disabled

Configuration Status:

SUCCESS (status updated 45 seconds ago)

DGMGRL>

1. Change the Data Guard protection mode to maximum performance.

DGMGRL> **edit configuration set protection mode as maxperformance;**

Succeeded.

1. Restart Redo Apply on the physical standby database and perform a log switch on the primary database.

DGMGRL> **edit database stndby set state = 'APPLY-ON';**

Succeeded.

DGMGRL> **SQL "alter system switch logfile";**

Succeeded.

1. Display the resulting configuration.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby - Physical standby database stndby2 - Logical standby database

Members Not Receiving Redo:

Fast-Start Failover: Disabled Configuration Status:

SUCCESS (status updated 60 seconds ago)

DGMGRL>

**Note:** It may take some time for Data Guard broker to resynchronize all the changes in this lab environment considering the hardware constraints. The following steps can be performed if needed:

**OPTIONAL LAB – do if we are waiting on other students to complete lab**

1. Before proceeding with additional lab steps, give the transport lag and apply lag an opportunity to catch up. Use the 'show configuration' and 'show database verbose stndby' commands until the lag clears. Repeat these commands as needed.

DGMGRL> **show database stndby**

Database - stndby

Role: PHYSICAL STANDBY

Intended State: APPLY-ON

Transport Lag: 0 seconds (computed 0 seconds ago) Apply Lag: 0 seconds (computed 0 seconds ago) Apply Rate: 0 Byte/s

Real Time Query: OFF Instance(s):

stndby

Database Status:

SUCCESS

1. Exit DGMGRL and SQL\*Plus leaving the terminal windows open for future practices.

# Practices for Lesson 13: Optimizing and Tuning a Data Guard Configuration

## Practices for Lesson 13: Overview

### Practices Overview

In these practices, you will configure network compression of redo data and SQL TUNING ADVISOR in a Data Guard environment.

## Practice 13-1: Configuring Network Compression of Redo Data

### Overview

In this practice, you will set the RedoCompression property to configure network compression of redo data.

### Tasks

1. Use a terminal window logged in as oracle to localhost with the environment variables set for orclcdb appropriately. Launch SQL\*Plus on your primary database and determine if redo compression is enabled by querying V$ARCHIVE\_DEST.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 12:32:07 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **col dest\_name format a30**

SQL> **select dest\_name, compression from v$archive\_dest;**

DEST\_NAME

COMPRES

LOG\_ARCHIVE\_DEST\_1 LOG\_ARCHIVE\_DEST\_2 LOG\_ARCHIVE\_DEST\_3 LOG\_ARCHIVE\_DEST\_4 LOG\_ARCHIVE\_DEST\_5 LOG\_ARCHIVE\_DEST\_6 LOG\_ARCHIVE\_DEST\_7 LOG\_ARCHIVE\_DEST\_8 LOG\_ARCHIVE\_DEST\_9

LOG\_ARCHIVE\_DEST\_10

DISABLE DISABLE DISABLE DISABLE DISABLE DISABLE DISABLE DISABLE DISABLE

DISABLE

Cd

LOG\_ARCHIVE\_DEST\_11

DISABLE

DEST\_NAME

COMPRES

LOG\_ARCHIVE\_DEST\_12 LOG\_ARCHIVE\_DEST\_13 LOG\_ARCHIVE\_DEST\_14 LOG\_ARCHIVE\_DEST\_15 LOG\_ARCHIVE\_DEST\_16 LOG\_ARCHIVE\_DEST\_17 LOG\_ARCHIVE\_DEST\_18 LOG\_ARCHIVE\_DEST\_19 LOG\_ARCHIVE\_DEST\_20 LOG\_ARCHIVE\_DEST\_21

LOG\_ARCHIVE\_DEST\_22

DISABLE DISABLE DISABLE DISABLE DISABLE DISABLE DISABLE DISABLE DISABLE DISABLE

DISABLE

DEST\_NAME

COMPRES

LOG\_ARCHIVE\_DEST\_23 LOG\_ARCHIVE\_DEST\_24 LOG\_ARCHIVE\_DEST\_25 LOG\_ARCHIVE\_DEST\_26 LOG\_ARCHIVE\_DEST\_27 LOG\_ARCHIVE\_DEST\_28 LOG\_ARCHIVE\_DEST\_29 LOG\_ARCHIVE\_DEST\_30

LOG\_ARCHIVE\_DEST\_31

DISABLE DISABLE DISABLE DISABLE DISABLE DISABLE DISABLE DISABLE

DISABLE

31 rows selected.

SQL>

1. Use a terminal window logged in as oracle to host02 with the environment variables set for orclcdbFS appropriately. Launch DGMGRL and connect to the primary database.

[oracle@host02 ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@host02 ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Fri Jun 5 12:34:02 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg/<password>@orclcdb** Connected to "orclcdb"

Connected as SYSDG.

DGMGRL>

1. Enable redo compression by setting the RedoCompression property for your database. Exit DGMGRL.

DGMGRL> **edit database 'orclcdb' set property 'RedoCompression'='ENABLE';**

Property "RedoCompression" updated

DGMGRL> **exitl**

[oracle@host02 ~]$

1. Return to your SQL\*Plus session on localhost and query V$ARCHIVE\_DEST again. Note that compression is set for LOG\_ARCHIVE\_DEST\_2. Exit SQL\*Plus.

SQL> **select dest\_name, compression from v$archive\_dest;**

DEST\_NAME

COMPRES

LOG\_ARCHIVE\_DEST\_1 LOG\_ARCHIVE\_DEST\_2 LOG\_ARCHIVE\_DEST\_3

…

DISABLE ENABLE

DISABLE

31 rows selected.

SQL> **show parameter log\_archive\_dest\_2**

NAME TYPE VALUE

log\_archive\_dest\_2 string

service="orclcdbfs",

SYNC AFFIRM delay=0

optional ***compression=enable*** max\_failure=0 max\_connections=1 reopen=300 db\_unique\_name="orclcdbFS" net\_timeout=30,

valid\_for=(online\_logfile

,all\_roles)

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

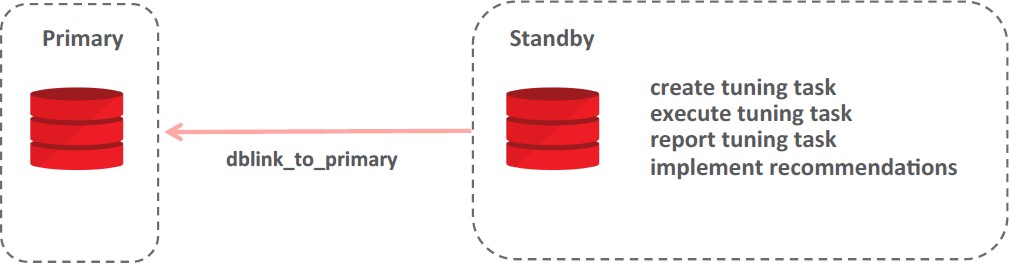
[oracle@localhost ~]$

## Practice 13-4: Using SQL Tuning Advisor for an Active Data Guard Instance

### Overview

In this practice, you will see how to use SQL Tuning Advisor for Active Data Guard.

The Active Data Guard (ADG) Databases are widely used to offload reporting or ad hoc query-only jobs from primary. Reporting workload profile is different from primary and often requires tuning. Starting with Oracle Database 12.2, you can run SQL Tuning Advisor to tune SQLs workloads running on ADG database.



* + All changes are done on primary and propagated from primary to standby by redo apply.
  + The data required for running the tuning tasks are fetched from the primary.
  + Support for PDB level tuning
  + Test execution (heavy lifting) happens on standby; only minimal write related activity on primary.

### Tasks

1. (**Reference Only, DO NOT RUN**) The environment for this practice has been set up with the setup\_STA.sh script. This script created the OE.PRODUCTS, OE.ORDER\_ITEMS, OE.orders, OE.CUSTOMERs, and OE.storeS tables.

[oracle@localhost ~]$ /home/oracle/setup/setup\_STA.sh

...

able OE.CUSTOMER:

30000 Rows successfully loaded.

Check the log file: control\_customer.log

for more information about the load. [oracle@localhost ~]$

1. Use a terminal window logged in as oracle to localhost with the environment variables set for orclcdb appropriately. Log in to the DEV1 PDB as the SYS user and run the setup13- 4.sql script.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base remains unchanged with value /u01/app/oracle

[oracle@localhost ~]$ **sqlplus sys/<password>@localhost:1521/dev1 as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 19:34:38 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **@/home/oracle/setup/setup13-4.sql**

SQL> exec dbms\_stats.delete\_table\_stats('OE','orders'); PL/SQL procedure successfully completed.

SQL> exec dbms\_stats.delete\_table\_stats('OE','order\_items); PL/SQL procedure successfully completed.

SQL>

1. Switch to the CDB root container and create a database link in the primary database for the standby database.

SQL> **connect / as sysdba**

Connected.

SQL> **CREATE DATABASE LINK dblink\_to\_primary CONNECT TO SYS$UMF IDENTIFIED BY <password> USING 'orclcdb';**

Database link created.

SQL>

1. Use a terminal window logged in as oracle to stndby with the environment variables set for stndby appropriately. Launch SQL\*Plus and run the problem query in the DEV1 PDB.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base remains unchanged with value /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 19:12:20 2020

Version 19.3.0.0.0

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

Last Successful login time: Fri Jun 05 2020 18:53:53 -04:00

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter session set container=dev1;**

Session altered.

SQL> **show pdbs**

CON\_ID CON\_NAME

OPEN MODE RESTRICTED

3 DEV1

READ ONLY NO

SQL> **@/home/oracle/setup/problem\_query.sql**

1. SQL> SELECT /\* problem\_query \*/
2. SUM(UNIT\_PRICE \*1.10) revenue, o.order\_id --, order\_status, order\_datetime
3. FROM Order\_items I join orders o
4. On o.order\_id = i.order\_id
5. WHERE o.order\_datetime < sysdate
6. group by o.order\_id

1950 rows

SQL>

1. Find sql\_id of the problem query.

SQL> **select sql\_id, sql\_text from v$sql where sql\_text like '%problem\_query%';**

SQL\_ID SQL\_TEXT

80rmhy60c1nga

select sql\_id, sql\_text from v$sql where sql\_text like '%problem\_query%'

an7zryzf86prm

SELECT /\* problem\_query \*/

SUM(UNIT\_PRICE \*1.10) revenue, o.order\_id --, order\_status, order\_datetime

FROM Order\_items I join orders o

On o.order\_id = i.order\_id

WHERE o.order\_datetime < sysdate

group by o.order\_id

SQL>

1. Open a new terminal window logged in as oracle to stndby with the environment variables set for stndby appropriately.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base has been set to /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Thu Jun 11 01:19:22 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL>

1. In the same terminal session, verify that the sql\_id of the problem query is visible. **Note:** At times, the problem query doesn’t appear in the CDB root container due to an internal delay. If that is the case, return to the terminal session used in step 4 and run the problem\_query.sql script again.

SQL> **select sql\_id, sql\_text from v$sql where sql\_text like '%problem\_query%';**

SQL\_ID SQL\_TEXT

80rmhy60c1nga

select sql\_id, sql\_text from v$sql where sql\_text like '%problem\_query%'

an7zryzf86prm

SELECT /\* problem\_query \*/

SUM(UNIT\_PRICE \*1.10) revenue, o.order\_id --, order\_status, order\_datetime

FROM Order\_items I join orders o

On o.order\_id = i.order\_id

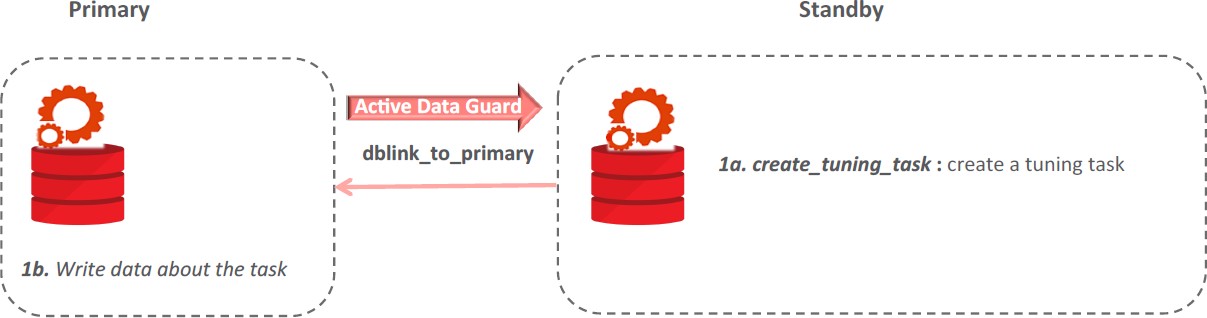
WHERE o.order\_datetime < sysdate

group by o.order\_id

SQL>

1. Create a SQL Tuning Task.

**Note:** If you receive ORA-13780: SQL statement does not exist, return to the terminal session used in step 4 and run the problem\_query.sql script again.



SQL> **@/home/oracle/setup/create\_sts.sql**

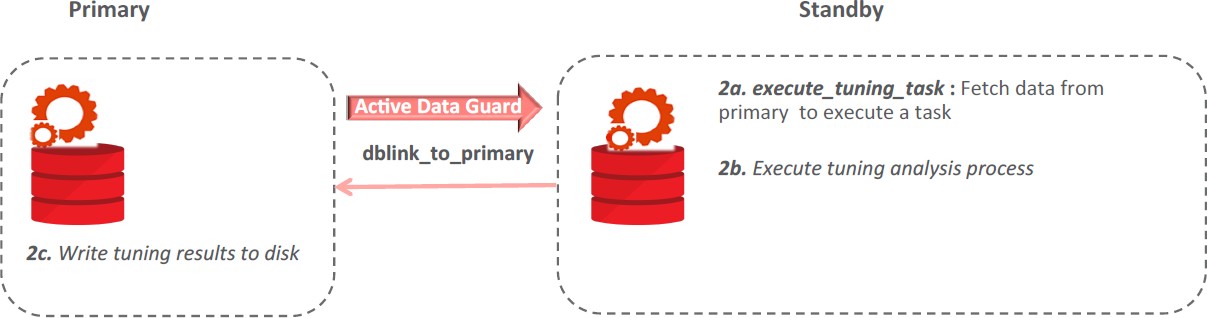
SQL> set echo on SQL> DECLARE

1. stmt\_task VARCHAR2(64);
2. BEGIN
3. stmt\_task:=dbms\_sqltune.create\_tuning\_task(sql\_id => 'an7zryzf86prm', task\_name => 'Tune\_problem\_query', database\_link\_to => 'DBLINK\_TO\_PRIMARY.EXAMPLE.COM');
4. END;

6 /

PL/SQL procedure successfully completed.

1. Execute the SQL Tuning Task.



SQL> **@/home/oracle/setup/exec\_sts.sql**

SQL> set echo on

SQL> EXECUTE dbms\_sqltune.execute\_tuning\_task(task\_name => 'Tune\_problem\_query');

PL/SQL procedure successfully completed.

SQL>

1. Generate the SQL Tuning Task report.

**Note:** The result varies.



SQL> **@/home/oracle/setup/get\_sts.sql**

SQL> SET linesize 200 SQL> SET LONG 999999999

SQL> SET pages 1000

SQL> SET longchunksize 20000

SQL> SELECT dbms\_sqltune.report\_tuning\_task('Tune\_problem\_query', 'TEXT', 'ALL') FROM dual;

DBMS\_SQLTUNE.REPORT\_TUNING\_TASK('TUNE\_PROBLEM\_QUERY','TEXT','ALL'

)

GENERAL INFORMATION SECTION

Tuning Task Name : Tune\_problem\_query Tuning Task Owner : OE

Tuning Task ID : 12

Workload Type : Single SQL Statement Execution Count : 1

Current Execution : EXEC\_22 Execution Type : TUNE SQL

Scope : COMPREHENSIVE

Time Limit(seconds): 1800 Completion Status : COMPLETED Started at : 06/05/2020 22:21:31

Completed at : 06/05/2020 22:21:35

Schema Name : OE Container Name: DEV1

SQL ID : an7zryzf86prm

SQL Text : SELECT /\* problem\_query \*/ SUM(lo\_extendedprice \* lo\_discount) revenue FROM oe.lineorder l, oe.date\_dim d

WHERE l.lo\_orderdate = d.d\_datekey

FINDINGS SECTION (2 findings)

* 1. Statistics Finding

Table "OE"."DATE\_DIM" was not analyzed.

Recommendation

* + - Consider collecting optimizer statistics for this table. execute dbms\_stats.gather\_table\_stats(ownname => 'OE',

tabname =>

'DATE\_DIM', estimate\_percent => DBMS\_STATS.AUTO\_SAMPLE\_SIZE,

method\_opt => 'FOR ALL COLUMNS SIZE AUTO');

Rationale

The optimizer requires up-to-date statistics for the table in order to select a good execution plan.

* 1. Statistics Finding

Table "OE"."order\_items" was not analyzed.

Recommendation

* + - Consider collecting optimizer statistics for this table. execute dbms\_stats.gather\_table\_stats(ownname => 'OE',

tabname =>

'ORDER\_ITEMS', estimate\_percent => DBMS\_STATS.AUTO\_SAMPLE\_SIZE,

method\_opt => 'FOR ALL COLUMNS SIZE AUTO');

Rationale

The optimizer requires up-to-date statistics for the table in order to

select a good execution plan.

EXPLAIN PLANS SECTION

1- Original

Plan hash value: 2963256899

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| | Id | | Operation | | | Name | | Rows | | Bytes | Cost (%CPU)| |
| Time | | |  |  |  |  |

| 0 | SELECT STATEMENT | | 1 | 52 | 6448

(1)| 00:00:01 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | 1 | | | SORT AGGREGATE  | | | |  | | 1 | | 52 | |  |
| |\* | 2 | | | HASH JOIN | | | | | 1610K| 79M| | 6448 | (1)| |

00:00:01 |

| 3 | TABLE ACCESS FULL| DATE\_DIM | 2556 | 33228 | 12

(0)| 00:00:01 |

| 4 | TABLE ACCESS FULL| LINEORDER | 1610K| 59M| 6431

(1)| 00:00:01 |

Query Block Name / Object Alias (identified by operation id):

1 - SEL$1

1. - SEL$1 / D@SEL$1
2. - SEL$1 / L@SEL$1

Predicate Information (identified by operation id):

2 - access("L"."LO\_ORDERDATE"="D"."D\_DATEKEY")

Column Projection Information (identified by operation id):

1. - (#keys=0) SUM("LO\_EXTENDEDPRICE"\*"LO\_DISCOUNT")[22]
2. - (#keys=1; rowset=256) "LO\_DISCOUNT"[NUMBER,22], "LO\_EXTENDEDPRICE"[NUMBER,22]
3. - (rowset=256) "D"."D\_DATEKEY"[NUMBER,22]
4. - (rowset=256) "L"."LO\_ORDERDATE"[NUMBER,22], "LO\_EXTENDEDPRICE"[NUMBER,22], "LO\_DISCOUNT"[NUMBER,22]

Note

- dynamic statistics used: dynamic sampling (level=2)

SQL>

1. Return to the terminal session connected to localhost. Switch to the DEV1 container.

SQL> **show con\_name**

CON\_NAME

CDB$ROOT SQL> SQL>

SQL> **alter session set container=dev1;**

Session altered.

SQL>

1. Optionally, implement the recommendations.

**Note:** If the recommendation is about the implementation of a profile, you can accept the profile directly in the standby database. The accepted profile is written to the primary database. Then the same profile is available in the standby database via redo apply.

SQL> **execute dbms\_stats.gather\_table\_stats(ownname => 'OE', tabname => 'ORDER\_ITEMS, estimate\_percent => DBMS\_STATS.AUTO\_SAMPLE\_SIZE, method\_opt => 'FOR ALL COLUMNS SIZE AUTO');**

PL/SQL procedure successfully completed.

SQL> **execute dbms\_stats.gather\_table\_stats(ownname => 'OE',tabname => 'ORDERS', estimate\_percent => DBMS\_STATS.AUTO\_SAMPLE\_SIZE, method\_opt => 'FOR ALL COLUMNS SIZE AUTO');**

PL/SQL procedure successfully completed. SQL>

1. Exit SQL\*Plus on all hosts leaving the current terminal windows.

# Practices for Lesson 14: Performing Role Transitions

## Practices for Lesson 14: Overview

### Practices Overview

In these practices, you will perform a switchover, and then switch back to the original configuration to observe the physical standby session connected during role transition.

## Practice 14-1: Performing Switchover

### Overview

In this practice, you will use DGMGRL view the configuration status, validate that the databases are ready for a role reversal, and then perform a switchover. During the switchover, you will observe how the session connected to the physical standby database is managed by default.

### Tasks

1. Use a terminal window on localhost connected as oracle with the environment variables set to orclcdb. Launch the DGMGRL utility and connect as the SYSDG user.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@localhost ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Sat Jun 6 07:29:43 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg/<password>@orclcdb** Connected to "orclcdb"

Connected as SYSDG.

DGMGRL>

1. Use the SHOW CONFIGURATION command to display the configuration status for the Data Guard configuration.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

orclcdb - Primary database instance

stndby - Physical standby database stndby2 - Logical standby database

Members Not Receiving Redo:

Fast-Start Failover: Disabled

Configuration Status:

SUCCESS (status updated 21 second ago)

DGMGRL>

1. Validate that the primary database is ready for role reversal using the VERBOSE option. The

VERBOSE option will show all checks being performed during validation.

DGMGRL> **validate database verbose orclcdb**

Database Role:

Primary database

Ready for Switchover: Yes

Flashback Database Status: orclcdb: Off

Capacity Information:

Database Instances Threads

orclcdb 1 1

Managed by Clusterware: orclcdb: NO

Validating static connect identifier for the primary database orclcdb...

The static connect identifier allows for a connection to database "orclcdb".

Temporary Tablespace File Information: orclcdb TEMP Files: 3

Data file Online Move in Progress: orclcdb: No

Transport-Related Information: Transport On: Yes

Log Files Cleared:

orclcdb Standby Redo Log Files: Cleared

DGMGRL>

1. Validate that the physical standby database is ready for role reversal using the VERBOSE

option.

#### DGMGRL> validate database verbose stndby

Database Role: Physical standby database Primary Database: orclcdb

Ready for Switchover: Yes

Ready for Failover: Yes (Primary Running)

Flashback Database Status: orclcdb: Off

stndby: Off

Capacity Information:

|  |  |  |
| --- | --- | --- |
| Database | Instances | Threads |
| orclcdb | 1 | 1 |
| stndby | 1 | 1 |

Managed by Clusterware: orclcdb: NO

stndby: NO

Validating static connect identifier for the primary database orclcdb...

The static connect identifier allows for a connection to database "orclcdb".

Temporary Tablespace File Information: orclcdb TEMP Files: 3

stndby TEMP Files: 3

Data file Online Move in Progress: orclcdb: No

stndby: No

Standby Apply-Related Information: Apply State: Running

Apply Lag: 0 seconds (computed 1 second ago) Apply Delay: 0 minutes

Transport-Related Information: Transport On: Yes

Gap Status: No Gap

Transport Lag: 0 seconds (computed 1 second ago) Transport Status: Success

Log Files Cleared:

orclcdb Standby Redo Log Files: Cleared stndby Online Redo Log Files: Cleared stndby Standby Redo Log Files: Available

Current Log File Groups Configuration:

Thread # Online Redo Log Groups Standby Redo Log Groups Status

(stndby)

1 3 2

Insufficient SRLs

Future Log File Groups Configuration:

Thread # Online Redo Log Groups Standby Redo Log Groups Status

(stndby)

1 3 0

Insufficient SRLs

Warning: standby redo logs not configured for thread 1 on orclcdb

Current Configuration Log File Sizes:

Thread # Smallest Online Redo Smallest Standby Redo Log File Size Log File Size

(stndby)

1 200 MBytes 200 MBytes

Apply-Related Property Settings:

Property orclcdb Value

stndby Value

|  |  |  |
| --- | --- | --- |
| DelayMins | 0 | 0 |
| ApplyParallel | AUTO | AUTO |
| ApplyInstances | 0 | 0 |

Transport-Related Property Settings:

Property orclcdb Value

stndby Value

LogShipping ON ON

LogXptMode ASYNC ASYNC

Dependency <empty> <empty>

DelayMins 0 0

|  |  |  |
| --- | --- | --- |
| Binding | optional | optional |
| MaxFailure | 0 | 0 |
| ReopenSecs | 300 | 300 |
| NetTimeout | 30 | 30 |
| RedoCompression | DISABLE | DISABLE |
| DGMGRL> | | |

1. Use the terminal connected to stndby as oracle with the environment variables set to

stndby. Launch SQL\*Plus and connect as the SYSDG user.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base has been set to /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sat Jun 6 07:36:51 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL>

1. Check the value of the STANDBY\_DB\_PRESERVE\_STATES parameter.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SQL> **show parameter STANDBY\_DB\_PRESERVE\_STATES** | | | | |
| NAME |  | TYPE |  | VALUE |
| standby\_db\_preserve\_states SQL> |  | string |  | NONE |

**Note:** When a physical standby database is converted to a primary, you have the option of keeping any sessions connected to the physical standby connected, without disruption,

during the switchover or failover. NONE means no sessions on the standby are retained during a switchover or failover. This is the default value.

1. Make sure that the real-time query is enabled in the physical standby database.

SQL> **select open\_mode from v$database;**

OPEN\_MODE

READ ONLY WITH APPLY SQL>

1. Return to the DGMGRL session on localhost. Switch over to the stndby physical standby database.

DGMGRL> **switchover to stndby**

Performing switchover NOW, please wait...

Operation requires a connection to database "stndby" Connecting ...

Connected to "stndby" Connected as SYSDG.

New primary database "stndby" is opening...

Operation requires start up of instance "orclcdb" on database "orclcdb"

Starting instance "orclcdb"... Connected to an idle instance. ORACLE instance started.

Connected to "orclcdb" Database mounted.

Database opened. Connected to "orclcdb"

Switchover succeeded, new primary is "stndby"

DGMGRL>

1. Display the new configuration.

**Note:** It takes a few minutes to clear up the ORA-\* error messages.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

stndby - Primary database stndbyFS - Far sync instance

orclcdb - Physical standby database stndby2 - Logical standby database

Members Not Receiving Redo: orclcdbFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:

SUCCESS (status updated 21 second ago)

DGMGRL> **exit**

[oracle@localhost ~]$

**Note:** Remember that the indentation used in the output of the SHOW CONFIGURATION

command indicates the hierarchy of how redo is being forwarded.

1. Return to the terminal session connected to stndby. Check the current status of the session that was connected to the original physical standby database. Exit SQL\*Plus.

SQL> **select open\_mode from v$database;**

select open\_mode from v$database

\*

ERROR at line 1:

ORA-03135: connection lost contact Process ID: 15166

Session ID: 52 Serial number: 2296

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0 [oracle@stndby ~]$

**Note:** The physical standby session was lost during role transition. This is the default behavior.

1. In the same terminal window on stndby, launch the DGMGRL utility and connect as the

SYSDG user.

[oracle@stndby ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Sat Jun 6 08:21:20 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg/<password>@stndby** Connected to "stndby"

Connected as SYSDG.

1. Perform a log switch on the new primary database stndby from within DGMGRL.

DGMGRL> **SQL "alter system switch logfile";**

Succeeded.

1. Verify that the new standby database orclcdb has zero transport lag and zero apply lag. You may need to wait a minute for this to clear.

DGMGRL> **show database orclcdb**

Database - orclcdb

Role:

Intended State: Transport Lag: Apply Lag:

PHYSICAL STANDBY APPLY-ON

0 seconds (computed 0 seconds ago)

0 seconds (computed 0 seconds ago)

Average Apply Rate: 2.00 KByte/s Real Time Query: ON Instance(s):

orclcdb

Database Status: SUCCESS

DGMGRL>

1. Exit DGMGRL leaving the terminal windows open.

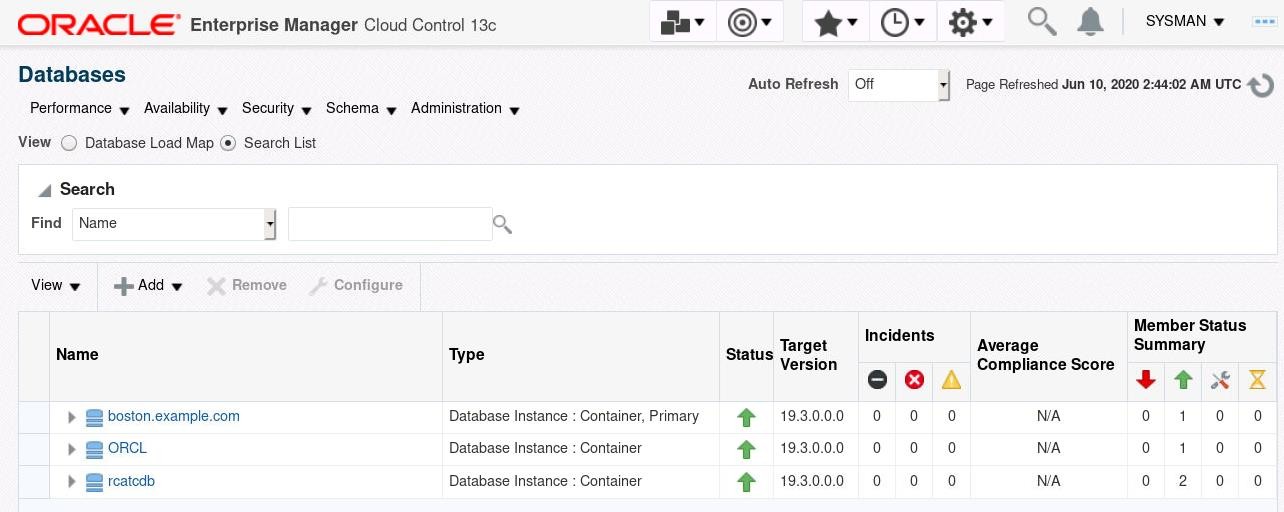
## Practice 14-2: Keeping Physical Standby Session Connected During Role Transition

### Overview

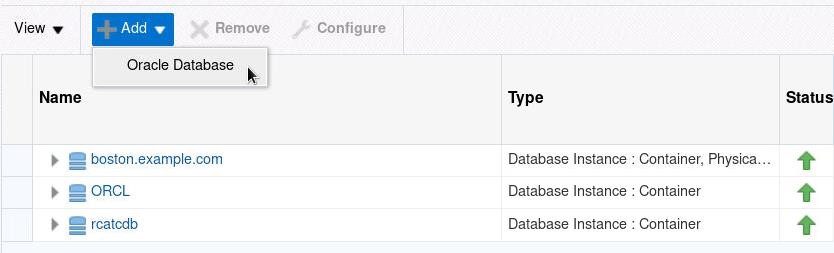
In this practice, you will use Enterprise Manager Cloud Control 13c and DGMGRL to view the configuration status, validate that the databases are ready for a role reversal, and then perform a switchover. During the switchover, you will observe how the session connected to the physical standby database is controlled with the STANDBY\_DB\_PRESERVE\_STATES parameter.

### Tasks

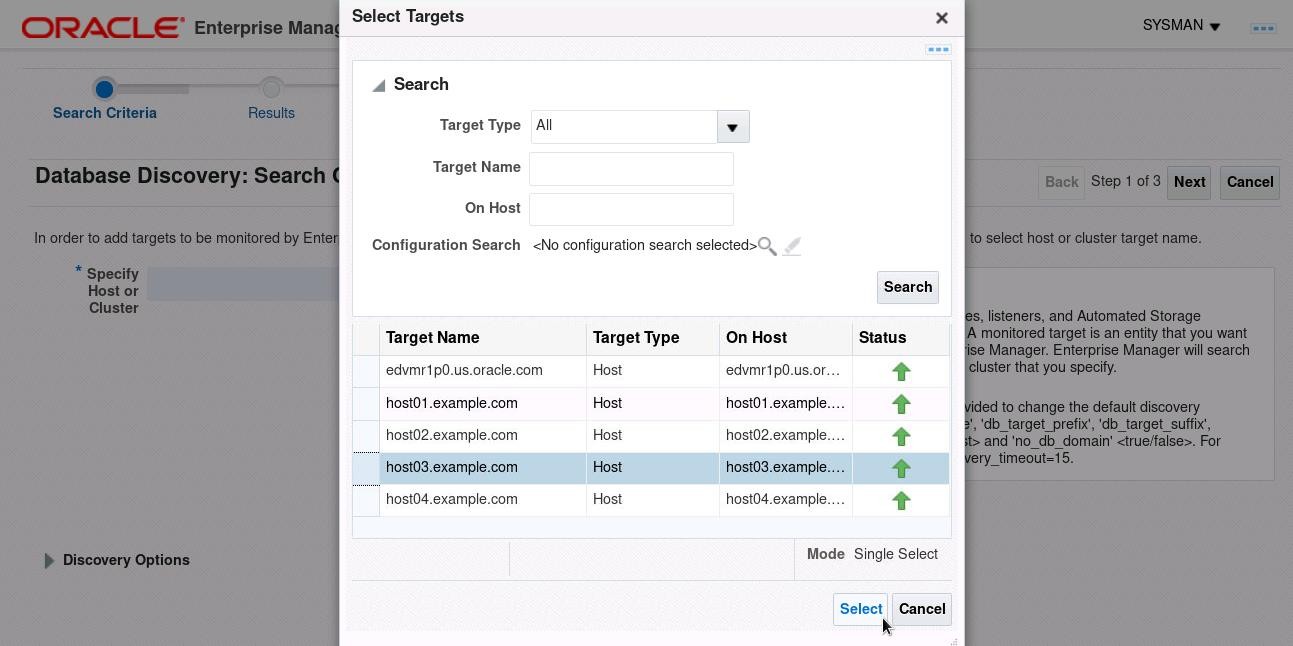
1. With the navigation techniques learned in practice 10-3, navigate to the Databases pages.



1. Add the stndby database as an EM target in preparation of the Switchover practice through Enterprise Manager. Click **Add** > **Oracle Database**.

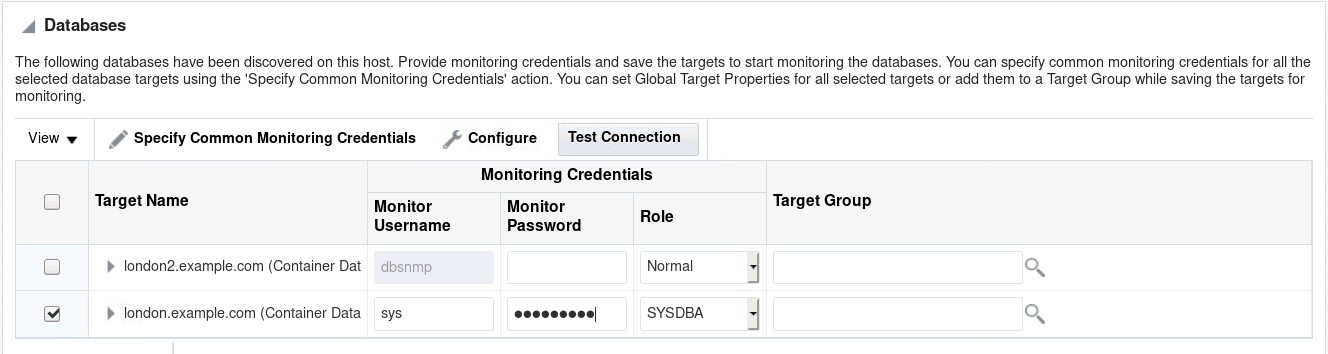


1. On the Database Discovery: Search Criteria page, enter the name of the host (stndby) by clicking the magnifying glass icon and selecting the host name in the dialog box. Then click **Next** to proceed.



1. On the Database Discovery: Results page, select the stndby database and provide the following information:
   * Monitor Username: sys
   * Monitor Password: <password>
   * Role: SYSDBA

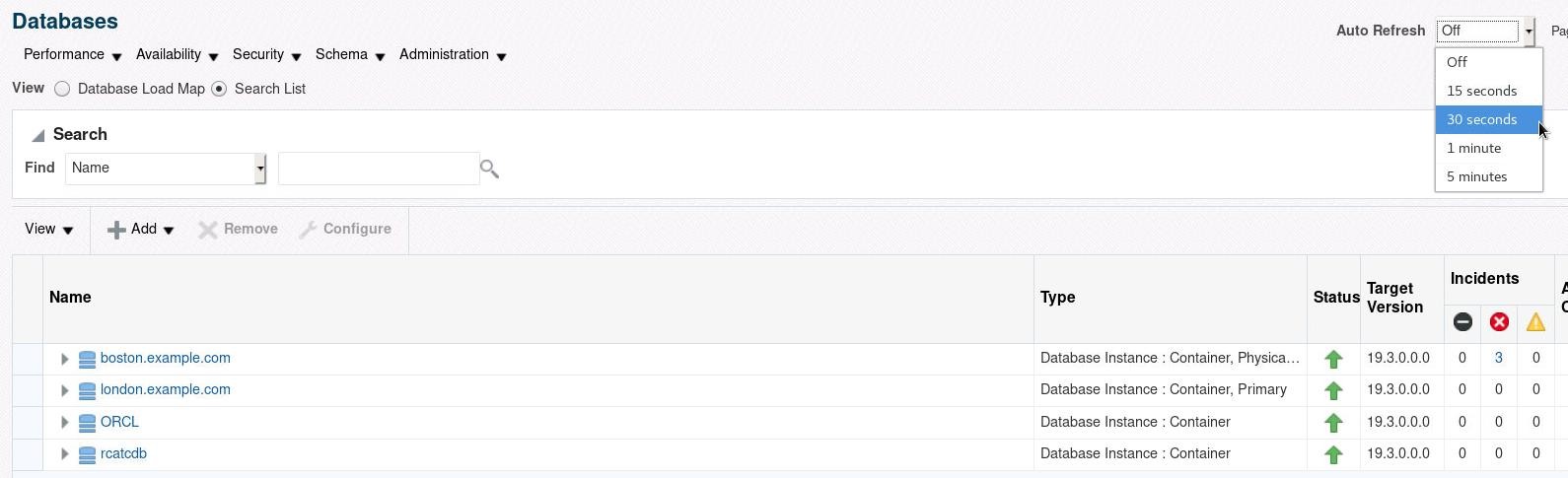
**Note:** You can also choose dbsnmp to lower the privilege instead of the SYS user.



1. Select the listed listener on stndby. Click **Next**.

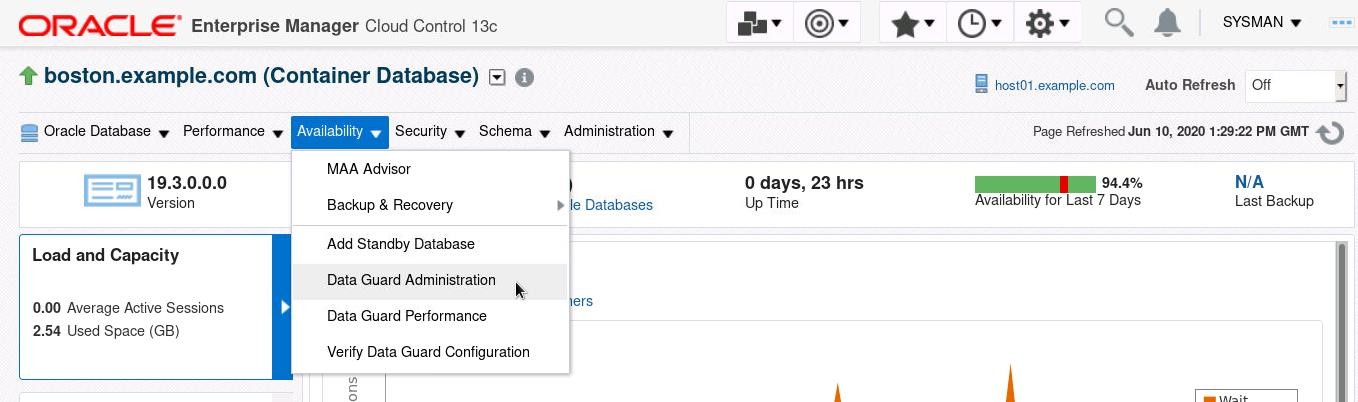


1. On the Database Discovery: Review page, click **Save**.
2. In the Confirmation dialog box, click **Close**.
3. On the Databases page, set the Auto Refresh option to 30 seconds and wait until the stndby.example.com target becomes normal. Click the link for the orclcdb.example.com target.

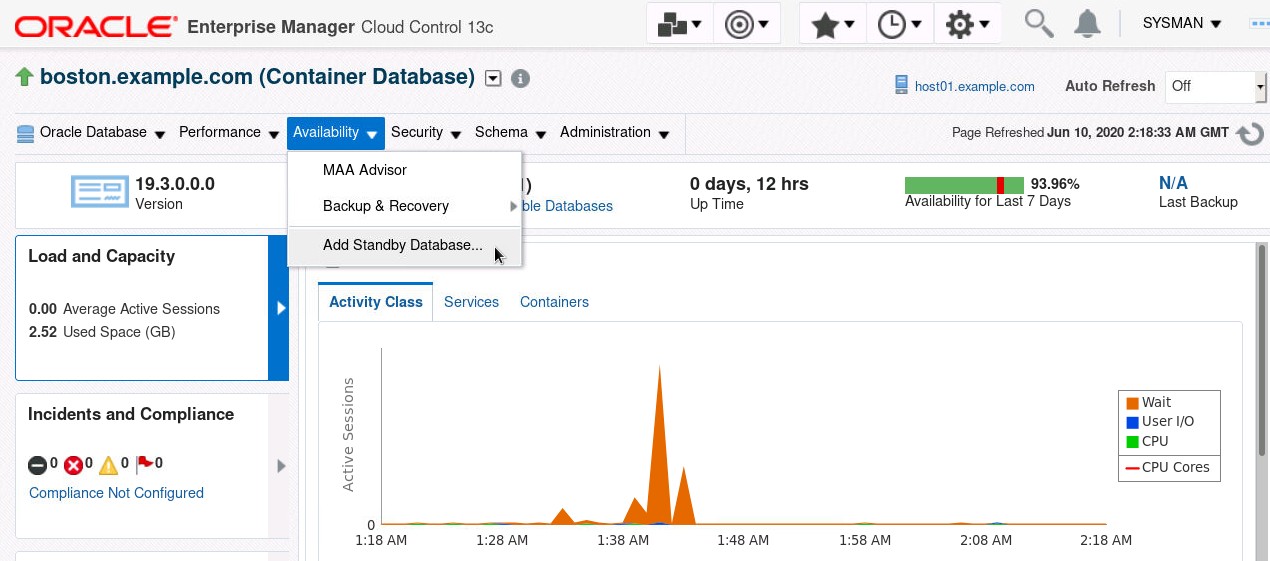


1. On the orclcdb.example.com database home page, select **Data Guard Administration**

from the Availability menu.

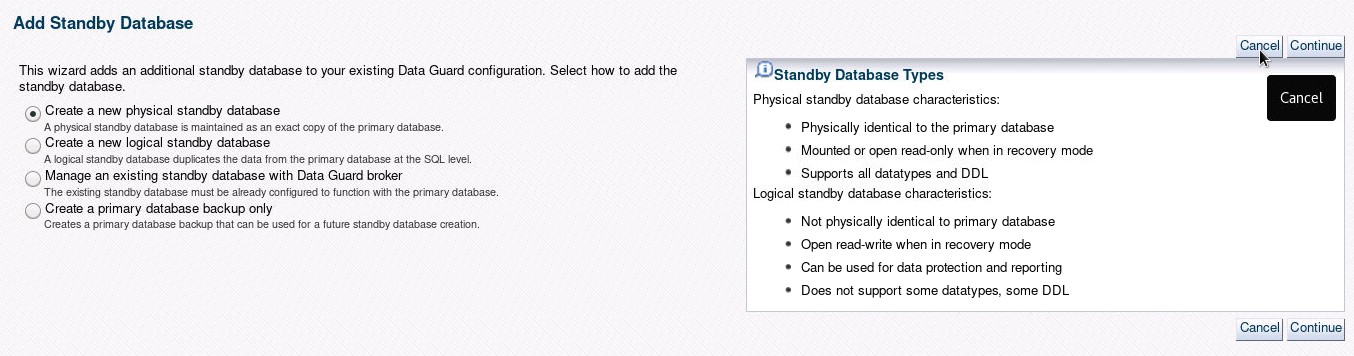


**(Optional)** If only the **Add Standby Database** link is visible, then select it. It will not launch the Add Standby Database Wizard, but instead, will navigate to the Data Guard home page.



**(Optional)** If the Add Standby Database link shows the Add Standby Database page, click

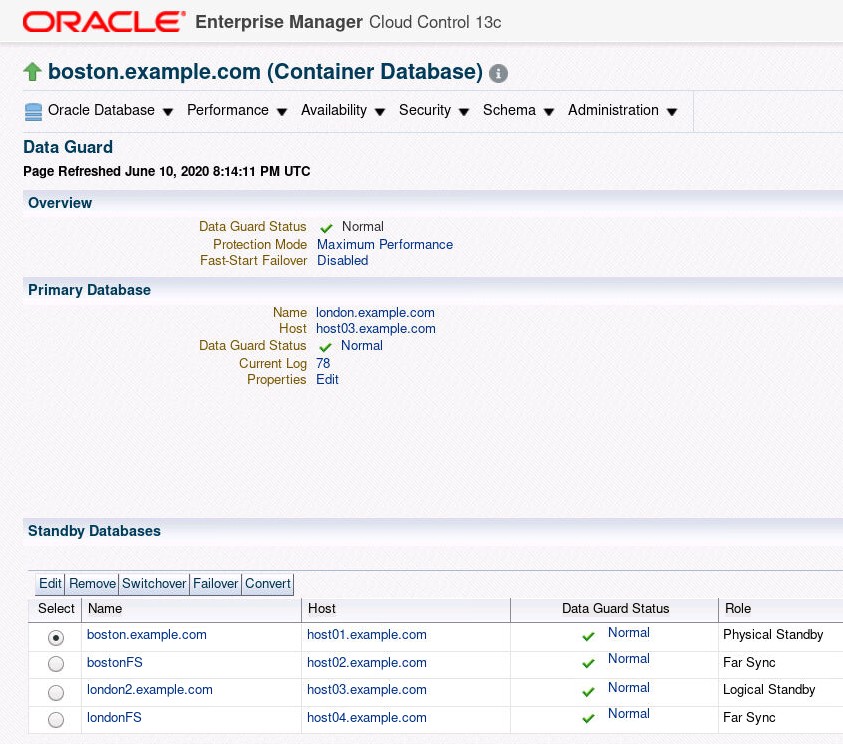
**Cancel** to navigate to the Data Guard home page.



(**Optional**) On the Database Login Page, select **New** in the Credential option with the following values. Click **Login**.

* + Username: sys
  + Password: *<password>*
  + Role: SYSDBA
  + Save As: NC\_ORCLCDB\_SYS2

1. On the Data Guard home page, make sure that the status of the current primary and physical standby is Normal.



1. Use the terminal connected to localhost as oracle with the environment variables set to

orclcdb. Launch SQL\*Plus and connect as the SYS user.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sat Jun 6 07:36:51 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL>

1. Set the value of the STANDBY\_DB\_PRESERVE\_STATES parameter to SESSION and restart the orclcdb database.

SQL> **alter system set STANDBY\_DB\_PRESERVE\_STATES = session**

**scope=spfile;**

System altered.

SQL> **shutdown immediate**

Database closed. Database dismounted.

ORACLE instance shut down. SQL> **startup**

ORACLE instance started.

Total System Global Area 629145352 bytes Fixed Size 9137928 bytes Variable Size 373293056 bytes Database Buffers 239075328 bytes

Redo Buffers 7639040 bytes Database mounted.

Database opened.

SQL> **show pdbs**

CON\_ID CON\_NAME

OPEN MODE RESTRICTED

1. PDB$SEED
2. DEV1

READ ONLY NO

MOUNTED

**Note:** When a physical standby database is converted to a primary, you have the option of keeping any sessions connected to the physical standby connected, without disruption,

during the switchover or failover. SESSION means user sessions are retained during a switchover or failover.

1. Open the DEV1 PDB and start the Media Recovery Process. Exit SQL\*Plus.

**Note:** If the Media Recovery process is already running, you will receive the ORA-01153

error message. You can safely proceed to the next step.

SQL> **alter pluggable database dev1 open;**

Pluggable database altered.

SQL> **alter database recover managed standby database disconnect;**

alter database recover managed standby database disconnect

\*

ERROR at line 1:

ORA-01153: an incompatible media recovery is active

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0 [oracle@localhost ~]$

1. Now, let’s establish a new session for testing.

[oracle@localhost ~]$ **sqlplus oe/<password>@localhost:1521/dev1.example.com**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sat Jun 6 08:57:31 2020

Version 19.3.0.0.0

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

Last Successful login time: Fri Jun 05 2020 22:19:26 -04:00 Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **col username format a10**

SQL> **select username, sid, serial# from v$session where sid=SYS\_CONTEXT('USERENV','SID');**

USERNAME

SID

SERIAL#

OE

42

40779

SQL>

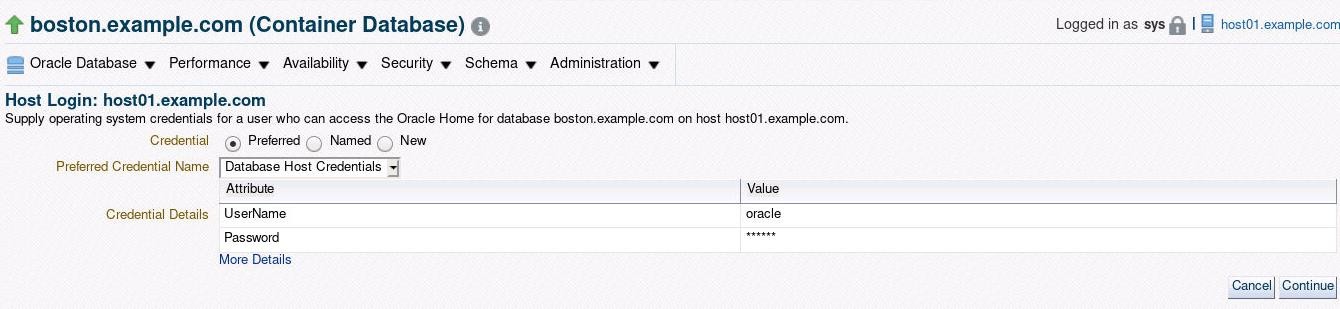
1. With the navigation techniques learned, return to the EM Data Guard Home page. Switch over to the orclcdb physical standby database like the following.

**Note:** If the EM page shows a warning message due to the restart of the orclcdb database, refresh the bowser.

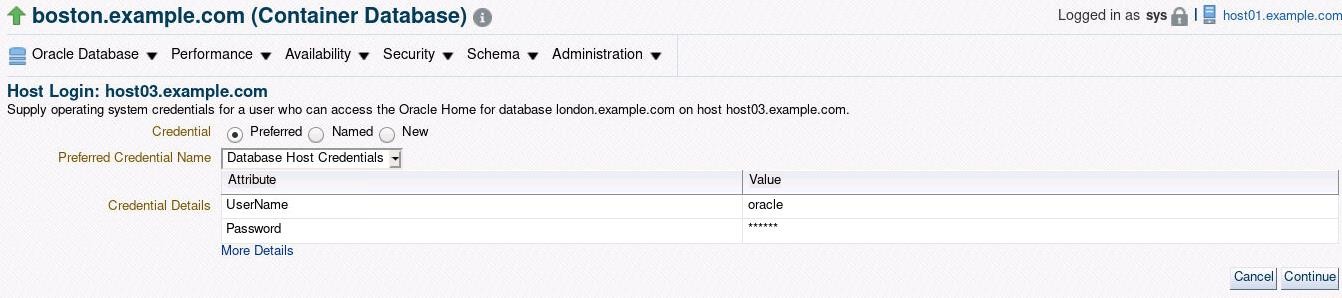
1. On the EM Data Guard Home page, select orclcdb.example.com in the Standby Databases section. Click **Switchover**.



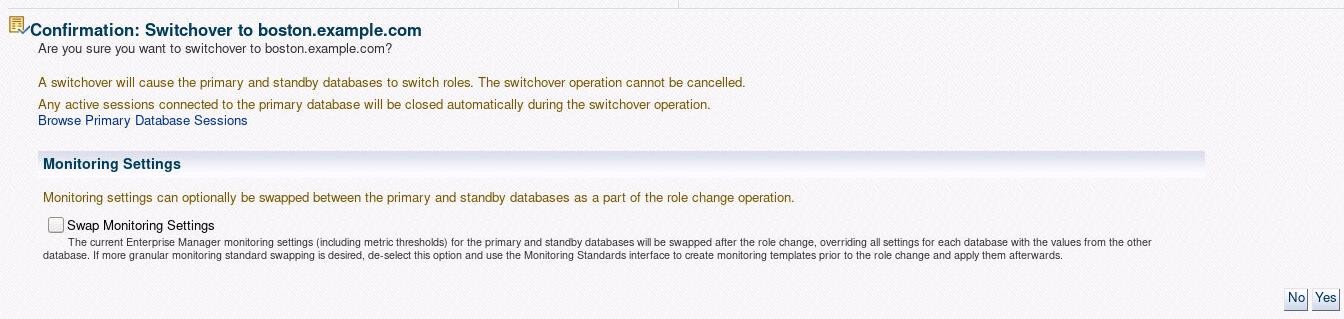
1. On the Host Login (localhost.example.com) page, select **Preferred** in the Credential option. Click **Continue**.



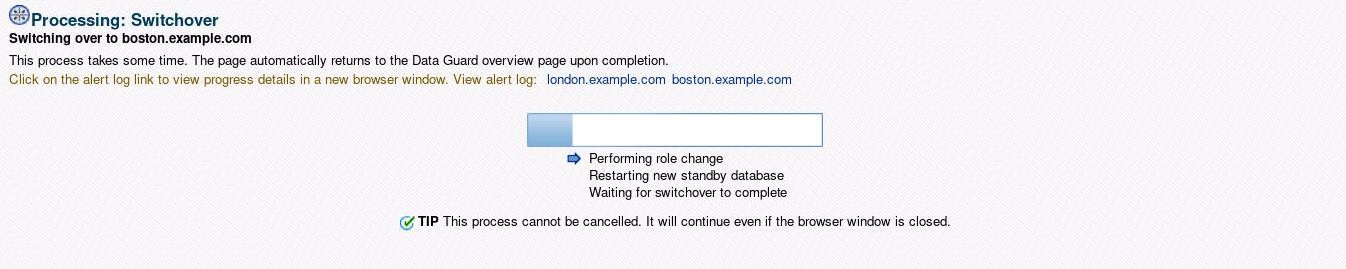
1. On the Host Login (stndby.example.com) page, select **Preferred** in the Credential option. Click **Continue**.



1. On the Confirmation page, click **Yes**.



1. Monitor the progress of Switchover. **DON’T WAIT** for completion. Move on to the next step.



1. Return to the SQL\*Plus session connected on localhost. Check the current status of the OE

session periodically. Exit SQL\*Plus.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SQL> **/**  USERNAME |  | SID |  | SERIAL# |  |
| OE  SQL> **/** |  |  |  | 42 | 40779 |
| USERNAME |  | SID |  | SERIAL# |  |
| OE  SQL> **/** |  |  |  | 42 | 40779 |

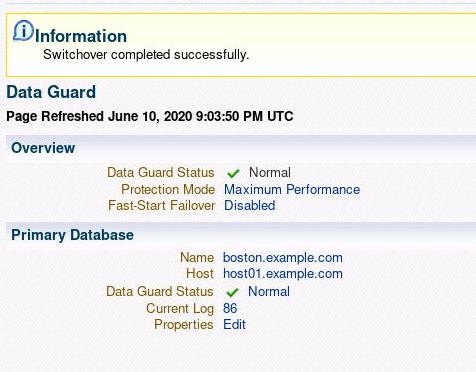
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| USERNAME |  | SID |  | SERIAL# |  |
| OE  SQL> **/** |  |  |  | 42 | 40779 |
| USERNAME |  | SID |  | SERIAL# |  |
| OE  SQL> **exit** |  |  |  | 42 | 40779 |

**Note:** The OE session hangs for a while and resumes. With the new feature, the session is retained during role transition.

Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0 [oracle@stndby ~]$

1. Return to the EM page. Once the switchover operation is complete, you will see the new primary database on the Data Guard home page.



# Practices for Lesson 15: Using Flashback Database in a Data Guard Configuration

## Practices for Lesson 15: Overview

### Practices Overview

In these practices, you will enable flashback database on both the primary database and the physical standby database. You will also test the automatic flashback of the physical standby database feature and manual flashback of the logical standby database.

## Practice 15-1: Configuring Flashback Database on the Primary Database

### Overview

In this practice, you will configure flashback database on the primary database and verify that it has been enabled.

### Tasks

1. Use a terminal window on localhost connected as oracle with the environment variables set to orclcdb. Launch SQL\*Plus and determine the current state of flashback database.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sat Jun 6 10:41:04 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **select flashback\_on from v$database;**

FLASHBACK\_ON NO

SQL>

1. Verify that the primary database is in archive log mode, a pre-requisite to flashback database.

SQL> **archive log list**

Database log mode Archive Mode

Automatic archival Enabled

Archive destination USE\_DB\_RECOVERY\_FILE\_DEST Oldest online log sequence 139

Next log sequence to archive 141

Current log sequence 141

SQL>

1. Verify that the fast recovery area has been configured for the primary database, a pre-requisite to flashback database.

|  |  |  |  |
| --- | --- | --- | --- |
| SQL> **show parameter db\_recovery**  NAME TYPE VALUE | | | |
| db\_recovery\_file\_dest  db\_recovery\_file\_dest\_size |  | string  big integer | /u01/app/oracle  /fast\_recovery\_area  15000M |

1. Determine the current amount of time in minutes for the flashback window.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SQL> **show parameter flashback**  NAME |  | TYPE |  | VALUE |
| db\_flashback\_retention\_target SQL> |  | integer |  | 1440 |

1. Adjust the flashback window to be 3 days (1440 minutes/day x 3 days = 4320 minutes).

SQL> **alter system set db\_flashback\_retention\_target = 4320;**

System altered.

1. Enable flashback database for the whole database.

SQL> **alter database flashback on;**

Database altered

1. Verify that flashback database has been enabled.

SQL> **select flashback\_on from v$database;**

FLASHBACK\_ON YES

1. Determine the current size (in bytes) of the flashback data.

SQL> **select flashback\_size from v$flashback\_database\_log;**

FLASHBACK\_SIZE

419430400

SQL>

1. Determine the name, quantity, and sizes of the flashback log files that were created when flashback database was enabled. Your file names will be different. Exit SQL\*Plus.

SQL> **select name,bytes from v$flashback\_database\_logfile;**

NAME

BYTES

/u01/app/oracle/fast\_recovery\_area/ORCLCDB/flashback/o1\_mf\_hfqbw2 0q\_.flb

209715200

/u01/app/oracle/fast\_recovery\_area/ORCLCDB/flashback/o1\_mf\_hfqbw9 o5\_.flb

209715200

SQL>

SQL> **exit;**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@localhost ~]$

## Practice 15-2: Configuring Flashback Database on the Physical Standby Database

### Overview

In this practice, you will enable flashback database on the physical standby database.

### Tasks

1. Use a terminal window on stndby connected as oracle with the environment variables set to stndby. Launch SQL\*Plus and determine the current state of the flashback database.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base has been set to /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sat Jun 6 10:45:35 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **select flashback\_on from v$database;**

FLASHBACK\_ON NO

SQL>

1. Verify that the physical standby database is in archive log mode, a pre-requisite to flashback database.

SQL> **archive log list** Database log mode Automatic archival

Archive destination

Archive Mode Enabled

USE\_DB\_RECOVERY\_FILE\_DEST

Oldest online log sequence 0

Next log sequence to archive 0

Current log sequence 0

SQL>

1. Verify that the fast recovery area has been configured for the physical standby database, a pre-requisite to flashback database.

|  |  |  |  |
| --- | --- | --- | --- |
| SQL> **show parameter db\_recovery**  NAME TYPE VALUE | | | |
| db\_recovery\_file\_dest  db\_recovery\_file\_dest\_size |  | string  big integer | /u01/app/oracle  /fast\_recovery\_area  15000M |

1. Determine the current amount of time in minutes for the flashback window.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SQL> **show parameter flashback** |  | | | |
| NAME |  | TYPE |  | VALUE |
| db\_flashback\_retention\_target |  | integer |  | 1440 |

1. Adjust the flashback window to be 3 days (1440 minutes/day x 3 days = 4320 minutes).

SQL> **alter system set db\_flashback\_retention\_target = 4320;**

System altered.

1. Enable flashback database for the whole database. Note the error message that is returned.

SQL> **alter database flashback on;**

alter database flashback on

\*

ERROR at line 1:

ORA-01153: an incompatible media recovery is active

1. Stop the managed recovery mode for the physical standby database.

SQL> **alter database recover managed standby database cancel;**

Database altered.

1. Return to the SQL\*Plus session on stndby connected to the stndby physical standby database and enable flashback database a second time.

SQL> **alter database flashback on;**

Database altered.

1. Verify that flashback database has been enabled.

SQL> **select flashback\_on from v$database;**

FLASHBACK\_ON YES

1. Restart the managed recovery mode for the stndby physical standby database. Exit SQL\*Plus when done.

SQL> **alter database recover managed standby database disconnect;**

Database altered. SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@stndby ~]$

## Practice 15-3: Configuring Flashback Database on the Logical Standby Database

### Overview

In this practice, you will enable flashback database on the logical standby database.

### Tasks

1. Use a terminal window on stndby2 connected as oracle with the environment variables set to stndby2. Launch SQL\*Plus and determine the current state of flashback database.

[oracle@stndby2 ~]$ **. oraenv**

ORACLE\_SID = [stndby] ? **stndby2**

The Oracle base remains unchanged with value /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sat Jun 6 10:47:39 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **select flashback\_on from v$database;**

FLASHBACK\_ON NO

SQL>

1. Verify that the logical standby database is in archive log mode, a pre-requisite to flashback database.

SQL> **archive log list**

Database log mode Archive Mode

Automatic archival Enabled

Archive destination USE\_DB\_RECOVERY\_FILE\_DEST Oldest online log sequence 31

Next log sequence to archive 33

Current log sequence 33

SQL>

1. Verify that the fast recovery area has been configured for the physical standby database, a pre-requisite to flashback database.

|  |  |  |  |
| --- | --- | --- | --- |
| SQL> **show parameter db\_recovery**  NAME TYPE VALUE | | | |
| db\_recovery\_file\_dest  db\_recovery\_file\_dest\_size |  | string  big integer | /u01/app/oracle  /fast\_recovery\_area  15000M |

1. Determine the current amount of time in minutes for the flashback window.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SQL> **show parameter flashback** |  | | | |
| NAME |  | TYPE |  | VALUE |
| db\_flashback\_retention\_target |  | integer |  | 1440 |

1. Adjust the flashback window to be 3 days (1440 minutes/day x 3 days = 4320 minutes).

SQL> **alter system set db\_flashback\_retention\_target = 4320;**

System altered.

1. Enable flashback database for the whole database. Note the error message that is returned.

SQL> **alter database flashback on;**

Database altered.

1. Verify that flashback database has been enabled.

SQL> **select flashback\_on from v$database;**

FLASHBACK\_ON YES

SQL> **exit;**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@stndby ~]$

## Practice 15-4: Testing Automatic Flashback of Standby Database

### Overview

In this practice, you will flash back your primary database after some incorrect updates to the database. After the primary database is recovered, you will observe the automatic flashback of standby database feature.

### Tasks

1. Use a terminal window on localhost connected as oracle with the environment variables set to orclcdb. Launch SQL\*Plus to connect to DEV1 PDB as the SYS user and create a guaranteed restore point called orclcdb\_grp.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [orclcdb] ? **orclcdb**

The Oracle base remains unchanged with value /u01/app/oracle [oracle@localhost ~]$ **sqlplus sys/<password>@localhost:1521/dev1 as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sat Jun 6 11:00:55 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **CREATE RESTORE POINT orclcdb\_grp GUARANTEE FLASHBACK DATABASE;**

Restore point created. SQL> **col name format a30**

SQL> **SELECT name, scn, replicated FROM v$restore\_point;**

NAME

SCN REP

ORCLCDB\_GRP

3955268 NO

1. View HR data to determine the sum of the SALARY column in the HR.EMPLOYEES table and the total number of employees in department 90. You will use this information for comparison during this practice.

SQL> **@/home/oracle/setup/view\_HR.sql**

SQL> SELECT sum(salary) FROM hr.employees;

SUM(SALARY)

691416

SQL> SELECT count(\*) FROM hr.employees where department\_id=90; COUNT(\*)

3

1. Execute the user\_errors.sql script to update tables in the HR schema. Assume that it creates issues from which you will “recover” by flashing back the database in this practice.

SQL> **@/home/oracle/setup/user\_errors.sql**

update hr.employees set department\_id = 90 where job\_id = 'IT\_PROG';

5 rows updated.

update hr.employees e set salary = least(e.salary,(select (min\_salary + max\_salary)/2 \* 1.10 from hr.jobs j where j.job\_id = e.job\_id)) where job\_id not like 'AD\_%';

103 rows updated.

COMMIT;

Commit complete SQL>

1. Query the updated data in the HR schema and compare the results to the values you received in the queries in step 2.

SQL> **@/home/oracle/setup/view\_HR.sql**

SQL> SELECT sum(salary) FROM hr.employees; SUM(SALARY)

679092.4

SQL> SELECT count(\*) FROM hr.employees where department\_id=90; COUNT(\*)

8

1. Use a terminal window on stndby connected as oracle with the environment variables set to stndby. Launch SQL\*Plus to connect to DEV1 PDB. If the DEV1 PDB is not open, open it first.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base has been set to /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sat Jun 6 15:50:36 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **show pdbs**

CON\_ID CON\_NAME

OPEN MODE RESTRICTED

1. PDB$SEED
2. DEV1

MOUNTED MOUNTED

*-- If the stndby database is in the MOUNT mode*

SQL> **alter database open;**

Database altered.

SQL> **alter pluggable database dev1 open;**

Pluggable database altered. SQL> **show pdbs**

CON\_ID CON\_NAME

OPEN MODE RESTRICTED

1. PDB$SEED
2. DEV1

READ ONLY NO

READ ONLY NO

SQL>

SQL> **alter session set container = DEV1;**

Session altered. SQL>

1. Verify that the restore points were replicated to the stndby standby database.

SQL> **col name format a30**

SQL> **SELECT name, scn, replicated FROM v$restore\_point;**

NAME

SCN REP

ORCLCDB\_GRP\_PRIMARY

3955268 YES

**Note:** The restore point created in the primary database was replicated to the physical standby database.

1. Query the data in the HR schema in the stndby physical standby database.

SQL> **@/home/oracle/setup/view\_HR.sql**

SQL> SELECT sum(salary) FROM hr.employees; SUM(SALARY)

679092.4

SQL> SELECT count(\*) FROM hr.employees where department\_id=90; COUNT(\*)

8

**Note:** As you can see, the unwanted changes were applied in the stndby physical standby database.

1. Return to the terminal session connected to localhost. Shut down and mount the orclcdb

database to prepare for the FLASHBACK DATABASE operation. Exit SQL\*Plus.

SQL> **connect / as sysdba**

Connected.

SQL> **shutdown immediate**

Database closed. Database dismounted.

ORACLE instance shut down. SQL> **startup mount**

ORACLE instance started.

Total System Global Area 629145352 bytes Fixed Size 9137928 bytes Variable Size 377487360 bytes Database Buffers 234881024 bytes

Redo Buffers 7639040 bytes Database mounted.

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0 [oracle@localhost ~]$

1. Log in to the RMAN utility to run the FLASHBACK DATABASE command to flash back the database to the restore point called orclcdb\_grp. Exit RMAN.

[oracle@localhost ~]$ **rman target "'/ as sysbackup'"**

Recovery Manager: Release 19.0.0.0.0 - Production on Sat Jun 6 19:27:13 2020

Version 19.3.0.0.0

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

connected to target database: ORCLCDB (DBID=2732274290, not open) RMAN> **FLASHBACK DATABASE TO RESTORE POINT ORCLCDB\_GRP;**

Starting flashback at 06-JUN-20

using target database control file instead of recovery catalog allocated channel: ORA\_DISK\_1

channel ORA\_DISK\_1: SID=15 device type=DISK

starting media recovery

media recovery complete, elapsed time: 00:00:03 Finished flashback at 06-JUN-20

RMAN> **exit**

Recovery Manager complete. [oracle@localhost ~]$

1. Using SQL\*Plus, log in as the SYS user to open the primary database and the DEV1 PDB in read-only mode.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sat Jun 6 19:30:34 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter database open read only;**

Database altered.

SQL> **alter pluggable database dev1 open read only;**

Pluggable database altered.

1. Connect to the DEV1 PDB to verify that the database was flashed back correctly by querying the HR.EMPLOYEES table again. The values should match the values you obtained in the queries in step 2.

SQL> **alter session set container = DEV1;**

Session altered.

SQL> **@/home/oracle/setup/view\_HR.sql**

SQL> SELECT sum(salary) FROM hr.employees; SUM(SALARY)

691416

SQL> SELECT count(\*) FROM hr.employees where department\_id=90; COUNT(\*)

3

**Note:** The flashback operation cleaned up the unwanted changes in the primary database.

1. Now, restart the primary database with RESETLOGS and make sure that the DEV1 PDB is open.

SQL> **connect / as sysdba**

Connected.

SQL> **shutdown immediate**

Database closed. Database dismounted.

ORACLE instance shut down. SQL> **startup mount**

ORACLE instance started.

Total System Global Area 629145352 bytes Fixed Size 9137928 bytes

Variable Size

377487360 bytes

Database Buffers 234881024 bytes

Redo Buffers 7639040 bytes Database mounted.

SQL> **alter database open resetlogs;**

Database altered. SQL> **show pdbs**

CON\_ID CON\_NAME OPEN MODE RESTRICTED

1. PDB$SEED
2. DEV1

READ ONLY NO READ WRITE NO

SQL>

1. Return to the terminal session connected to stndby to check if the automatic flashback feature was used in the stndby standby database.

SQL> **@/home/oracle/setup/view\_HR.sql**

SQL> SELECT sum(salary) FROM hr.employees; SUM(SALARY)

679092.4

SQL> SELECT count(\*) FROM hr.employees where department\_id=90; COUNT(\*)

8

**Note:** The physical standby database still shows the unwanted changes.

1. In the same SQL\*Plus session on stndby, review the alert log file. Press CTRL+C to exit.

SQL> **host tail -100**

**/u01/app/oracle/diag/rdbms/stndby/stndby/trace/alert\_stndby.log|mo re**

...

Errors in file

/u01/app/oracle/diag/rdbms/stndby/stndby/trace/stndby\_mrp0\_26165.t rc:

ORA-19909: datafile 1 belongs to an orphan incarnation ORA-01110: data file 1: '/u01/app/oracle/oradata/STNDBY/system01.dbf'

2020-06-06T20:36:13.138213-04:00

MRP0 (PID:26165): Recovery coordinator encountered one or more errors during automatic flashback on standby

2020-06-06T20:36:13.138312-04:00

Background Media Recovery process shutdown (stndby)

2020-06-06T20:37:17.941175-04:00

rfs (PID:26865): Opened log for T-1.S-1 dbid 2732274290 branch 1042403723

2020-06-06T20:37:17.947674-04:00

rfs (PID:26865): Archived Log entry 5 added for B-1042403723.T- 1.S-1 ID 0xa2e0186f LAD:2

2020-06-06T20:37:18.004013-04:00

rfs (PID:26867): Opened log for T-1.S-2 dbid 2732274290 branch 1042403723

2020-06-06T20:37:18.008815-04:00

rfs (PID:26867): Archived Log entry 6 added for B-1042403723.T- 1.S-2 ID 0xa2e0186f LAD:2

2020-06-06T20:37:43.939464-04:00

Control autobackup written to DISK device

handle '/u01/app/oracle/fast\_recovery\_area/STNDBY/autobackup/2020\_06\_06/o 1\_mf\_s\_1042403484\_hfrfpq8w\_.bkp'

*==== CTRL + C =====*

SQL>

**Note:** The alert log indicates that the automatic flashback on standby didn’t work because the standby database is currently open. This feature works in the MOUNT state.

1. Connect as the SYS user and mount the stndby database.

SQL> **connect / as sysdba**

Connected.

SQL> **shutdown immediate**

Database closed. Database dismounted.

ORACLE instance shut down. SQL> **startup mount**

ORACLE instance started.

Total System Global Area 880802384 bytes

Fixed Size Variable Size Database Buffers Redo Buffers Database mounted. SQL>

9140816 bytes

767557632 bytes

96468992 bytes

7634944 bytes

1. Review the alert log file again. Press Ctrl+C to exit.

SQL> **host grep -i flashback**

**/u01/app/oracle/diag/rdbms/stndby/stndby/trace/alert\_stndby.log|mo re**

...

Flashback Restore Start

Flashback Restore Complete Flashback Media Recovery Start Flashback Media Recovery Complete

...

*==== CTRL + C =====*

SQL>

**Note:** The alert log file includes the Flashback Media Recovery Complete message, which indicates the automatic flashback of physical standby feature was applied in the stndby physical standby database.

1. Open the standby database and DEV1 PDB.

SQL> **alter database open;**

Database altered.

SQL> **alter pluggable database dev1 open;**

Pluggable database altered. SQL>

1. Run the same query to see if the unwanted changes were cleaned up.

SQL> **alter session set container = DEV1;**

Session altered.

SQL> **@/home/oracle/setup/view\_HR.sql**

SQL> SELECT sum(salary) FROM hr.employees; SUM(SALARY)

691416

SQL> SELECT count(\*) FROM hr.employees where department\_id=90; COUNT(\*)

3

SQL>

**Note:** The flashback operation was automatically performed in the physical standby database. So, the unwanted changes were cleaned up.

1. Return to the SQL\*Plus session on localhost connected to the primary database. Switch to the DEV1 PDB and drop the restore point.

SQL> **alter session set container = DEV1;**

Session altered.

SQL> **drop restore point orclcdb\_grp;**

Restore point dropped. SQL>

1. Exit SQL\*Plus on localhost and stndby leaving the terminal windows open for future practices.

## Practice 15-5: Performing Flashback of the Logical Standby Database

### Overview

In this practice, you will examine the stndby2 logical standby database to test if the automatic flashback of standby database feature works with the logical standby database or not. Finally, you will perform the flashback of the stndby2 logical standby database manually.

### Tasks

1. Use the terminal window connected to stndby as oracle with the environment variables set to stndby2. Connect to the DEV1 PDB.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby2**

The Oracle base remains unchanged with value /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sat Jun 6 13:13:37 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter session set container=dev1;**

Session altered.

SQL>

1. Query the updated data in the HR schema.

SQL> **@/home/oracle/setup/view\_HR.sql**

SQL> SELECT sum(salary) FROM hr.employees; SUM(SALARY)

679092.4

SQL> SELECT count(\*) FROM hr.employees where department\_id=90; COUNT(\*)

8

SQL>

1. Open a new terminal window connected to stndby. Review the alert log file for the

stndby2 standby database.

[oracle@stndby ~]$ **tail -f**

**/u01/app/oracle/diag/rdbms/stndby2/stndby2/trace/alert\_stndby2.log**

LOGMINER: Memory Release Limit: 1M LOGMINER: Max Decomp Region Memory: 1M LOGMINER: Transaction Queue Size: 1024 2020-06-06T13:05:34.735266-04:00

Fatal Error: LogMiner: session# 1 processed beyond new branch scn.

LOGSTDBY status: ORA-01346: Oracle LogMiner processed redo beyond primary reset log SCN 3878101

2020-06-06T13:05:34.738051-04:00

Errors in file

/u01/app/oracle/diag/rdbms/stndby2/stndby2/trace/stndby2\_lsp0\_9101

.trc:

ORA-01346: Oracle LogMiner processed redo beyond primary reset log SCN 3878101

**Note:** The alert log file shows the automatic flashback of the logical standby database didn’t work.

1. Since the automatic flashback of standby feature works for the physical standby database, you will have to flash back the logical standby database manually. Use the terminal window to connect to localhost as oracle.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [orclcdb] ? **orclcdb**

The Oracle base remains unchanged with value /u01/app/oracle [oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sat Jun 6 13:23:11 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL>

1. Using SQL\*Plus, determine an SCN that is at least two SCNs prior to the SCN when the OPEN RESETLOGS command was issued. This is necessary to enable the standby to recover properly through OPEN RESETLOGS. Use the following query to find the “before RESETLOGS” SCN

SQL> **SELECT TO\_CHAR(resetlogs\_change# - 2) FROM v$database;**

TO\_CHAR(RESETLOGS\_CHANGE#-2) 3955268

SQL>

1. Return the SQL\*Plus session connected to the stndby2 database. Determine the target SCN for flashback operation at the logical standby. In this step, the FLASHBACK\_SCN value for PRIMARY\_SCN is from Step 5.

SQL> **connect / as sysdba**

Connected.

SQL> **SELECT DBMS\_LOGSTDBY.MAP\_PRIMARY\_SCN(PRIMARY\_SCN => 3955268) AS TARGET\_SCN from DUAL;**

TARGET\_SCN 3598897

SQL>

1. Flash back the standby database to the “before RESETLOGS” SCN that you queried in step 6.

**Note:** Your SCN is different.

SQL> **shutdown immediate**

Database closed. Database dismounted.

ORACLE instance shut down. SQL> **startup mount exclusive** ORACLE instance started.

Total System Global Area 880802384 bytes

Fixed Size Variable Size Database Buffers Redo Buffers

9140816 bytes

767557632 bytes

96468992 bytes

7634944 bytes

Database mounted.

SQL> **FLASHBACK DATABASE TO SCN *<SCN in step 6>*;**

Flashback complete.

1. Open the stndby2 database in READ ONLY mode and verify the HR data in the DEV1 PDB.

SQL> **alter database open read only;**

Database altered.

SQL> **alter pluggable database dev1 open read only;**

Pluggable database altered.

SQL> **alter session set container=DEV1;**

Session altered.

SQL> **@/home/oracle/setup/view\_HR.sql**

SQL> SELECT sum(salary) FROM hr.employees; SUM(SALARY)

691416

SQL> SELECT count(\*) FROM hr.employees where department\_id=90; COUNT(\*)

3

SQL>

**Note:** As you can see, the logical standby database was successfully flashed back.

1. Open the stndby2 database with RESETLOGS.

SQL> **connect / as sysdba**

Connected.

SQL> **shutdown immediate**

Database closed. Database dismounted.

ORACLE instance shut down. SQL> **startup mount**

ORACLE instance started.

Total System Global Area 880802384 bytes Fixed Size 9140816 bytes Variable Size 658505728 bytes

Database Buffers 205520896 bytes

Redo Buffers 7634944 bytes Database mounted.

SQL> **ALTER DATABASE OPEN RESETLOGS;**

Database altered.

SQL> **ALTER PLUGGABLE DATABASE DEV1 OPEN;**

Pluggable database altered.

1. Restart SQL Apply on the standby database if it’s not running. The standby database will be ready to receive and apply logs from the primary database.

**Note:** If SQL Apply is already running, you will receive ORA-16103. Exit SQL\*Plus.

SQL> **ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;**

Database altered. SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0 [oracle@stndby ~]$

1. Launch the DGMGRL utility and connect as the SYSDG user.

[oracle@stndby ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Sat Jun 6 10:51:51 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg/<password>@orclcdb** Connected to "orclcdb"

Connected as SYSDG.

1. Use the SHOW CONFIGURATION command to display the configuration status for the Data Guard configuration.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby - Physical standby database stndby2 - Logical standby database

Members Not Receiving Redo: stndbyFS - Far sync instance

Fast-Start Failover: Disabled Configuration Status:

SUCCESS (status updated 56 seconds ago)

DGMGRL>

1. Exit DGMGRL and SQL\*Plus leaving the terminal windows open for future practices.

# Practices for Lesson 16: Enabling Fast-Start Failover

## Practices for Lesson 16: Overview

### Practices Overview

In these practices, you will set up and configure fast-start failover. You will then simulate a failure of the primary database and observe the automatic failover to the standby database.

## Practice 16-1: Configuring Fast-Start Failover in Observer-Only Mode

### Overview

In this practice, you will configure fast-start failover in observe-only mode. After configuring fast- start failover in observe-only mode, you will start the observer process. Then you will simulate the crash of the primary database for FSFO dry-run.

### Tasks

1. Use a terminal window on localhost connected as oracle with the environment variables set to orclcdb. Launch SQL\*Plus and perform a log switch on the primary database. Exit SQL\*Plus.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@localhost ~]$

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sat Jun 6 23:13:40 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter system switch logfile;**

System altered.

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@localhost ~]$

1. Use a terminal window on connected as oracle with the environment variables set to stndby. Launch the DGMGRL utility and connect as the SYSDG user.

[oracle@host02 ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@host02 ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Sat Jun 6 23:17:05 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg/<password>@stndby** Connected to "stndby"

Connected as SYSDG.

DGMGRL>

1. Verify that there is no Transport Lag or Apply lag at the physical standby database and logical standby database before proceeding with labs.

DGMGRL> **show database stndby**

Database - stndby

Enterprise Manager Name: stndby: 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: 7.00 KByte/s

Real Time Query: ON Instance(s):

stndby

Database Status:

SUCCESS

DGMGRL> **show database stndby2**

Database - stndby2

Enterprise Manager Name: stndby2

Role: LOGICAL STANDBY

Intended State: APPLY-ON

Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Active Apply Rate: 785.02 KByte/s

Instance(s): stndby2

Database Status:

SUCCESS

sho

1. Display the current configuration and note the current state of fast-start failover.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

stndby - Primary database instance

stndby - Physical standby database stndby2 - Logical standby database

Members Not Receiving Redo:

Fast-Start Failover: Disabled

Configuration Status:

SUCCESS (status updated 46 seconds ago)

DGMGRL>

1. Set up the FastStartFailoverTarget configuration property on the orclcdb primary database to indicate the desired stndby target standby database.

DGMGRL> **edit database orclcdb set property FastStartFailoverTarget = stndby;**

Property "faststartfailovertarget" updated

1. Define the reciprocal fast-start failover target for when the stndby physical standby database becomes the primary database. This would be set automatically by the broker if far sync was not in the configuration.

DGMGRL> **edit database stndby set property FastStartFailoverTarget = orclcdb;**

Property "faststartfailovertarget" updated

1. Display the FastStartFailoverTarget configuration property for both the primary database and the physical standby database.

DGMGRL> **show database stndby faststartfailovertarget;**

FastStartFailoverTarget = 'orclcdb'

DGMGRL> **show database orclcdb FastStartFailoverTarget;**

FastStartFailoverTarget = 'stndby'

1. Modify the ObserverReconnect configuration property and set the value to 120 seconds.

DGMGRL> **edit configuration set property ObserverReconnect=120;**

Property " observerreconnect" updated

1. To enable fast-start failover the configuration must be upgraded to the maximum availability mode. Display the current RedoRoutes property for both the primary and standby database.

DGMGRL> **show database orclcdb redoroutes;**

RedoRoutes = '(orclcdb:orclcdb SYNC)' DGMGRL> **show database stndby redoroutes;**

RedoRoutes = '(stndby:stndby SYNC)'

If this is blank do this:

DGMGRL> edit database stndby set property 'RedoRoutes'='(stndby:orclcdb ASYNC)';

**Note:** FASTSYNC would also be acceptable settings for the maximum availability.

1. Upgrade the protection mode to maximum availability.

DGMGRL> **edit configuration set protection mode as maxavailability;**

Succeeded. –If this fails continue

1. Configure fast-start failover in observe-only mode to test how fast-failover will work in your environment.

DGMGRL> **enable fast\_start failover observe only;**

Enabled in Observe-Only Mode. If this fails – continue on.

1. Start the observer process.

DGMGRL> **start observer**

[W000 2020-06-06T23:31:17.857-04:00] FSFO target standby is

stndby

Observer started

[W000 2020-06-06T23:31:18.273-04:00] Observer trace level is set to USER

[W000 2020-06-06T23:31:18.273-04:00] Try to connect to the primary.

[W000 2020-06-06T23:31:18.273-04:00] Try to connect to the primary orclcdb.

[W000 2020-06-06T23:31:18.314-04:00] The standby stndby is ready to be a FSFO target

[W000 2020-06-06T23:31:18.314-04:00] Reconnect interval expired, create new connection to primary database.

[W000 2020-06-06T23:31:18.314-04:00] Try to connect to the primary.

[W000 2020-06-06T23:31:18.375-04:00] Connection to the primary restored!

[W000 2020-06-06T23:31:24.394-04:00] Disconnecting from database orclcdb.

**Note:** The prompt will not return after starting the observer unless you start the observer in the background mode. Keep this terminal window open with the observer running in it.

1. Return to the terminal session connected to localhost. Simulate a crash of the primary database.

[oracle@localhost ~]$ **pgrep -lf smon**

21521 ora\_smon\_stndby

[oracle@localhost ~]$ **kill -9 21521 or whatever smon for stndby is**

[oracle@localhost ~]$

1. Return to the Observer session Review the output.

DGMGRL> **start observer**

...

[W000 2020-06-06T23:33:18.179-04:00] Try to connect to the primary.

[W000 2020-06-06T23:34:48.972-04:00] Primary database cannot be reached.

[W000 2020-06-06T23:34:48.972-04:00] Fast-Start Failover threshold has not exceeded. **Retry for the next 30 seconds**

[W000 2020-06-06T23:34:49.973-04:00] Try to connect to the primary.

ORA-12537: TNS:connection closed

Unable to connect to database using orclcdb

[W000 2020-06-06T23:35:15.727-04:00] Primary database cannot be reached.

[W000 2020-06-06T23:35:15.727-04:00] Fast-Start Failover threshold has not exceeded. **Retry for the next 3 seconds**

[W000 2020-06-06T23:35:16.727-04:00] Try to connect to the primary.

[W000 2020-06-06T23:35:17.806-04:00] Primary database cannot be reached.

[W000 2020-06-06T23:35:17.806-04:00] Fast-Start Failover threshold has not exceeded. **Retry for the next 1 second**

[W000 2020-06-06T23:35:18.806-04:00] Try to connect to the primary.

[W000 2020-06-06T23:35:20.000-04:00] Primary database cannot be reached.

[W000 2020-06-06T23:35:20.000-04:00] **Fast-Start Failover threshold has expired.**

[W000 2020-06-06T23:35:20.000-04:00] Try to connect to the standby.

[W000 2020-06-06T23:35:20.000-04:00] Making a last connection attempt to primary database before proceeding with Fast-Start Failover.

[W000 2020-06-06T23:35:20.000-04:00] Check if the standby is ready for failover.

[W000 2020-06-06T23:35:20.005-04:00] A fast-start failover would have been initiated...

[W000 2020-06-06T23:35:20.005-04:00] **Unable to failover since this observer is in observe-only mode**

[W000 2020-06-06T23:35:20.005-04:00] **Fast-Start Failover is not possible because observe-only mode.**

[W000 2020-06-06T23:35:21.006-04:00] Try to connect to the primary.

[W000 2020-06-06T23:35:22.095-04:00] Primary database cannot be reached.

[W000 2020-06-06T23:35:23.096-04:00] Try to connect to the primary.

[W000 2020-06-06T23:35:24.214-04:00] Primary database cannot be reached.

[W000 2020-06-06T23:35:25.215-04:00] Try to connect to the primary.

[W000 2020-06-06T23:35:49.321-04:00] Primary database cannot be reached.

[W000 2020-06-06T23:35:49.321-04:00] Fast-Start Failover threshold has not exceeded. Retry for the next 2 seconds

[W000 2020-06-06T23:35:50.322-04:00] Try to connect to the primary.

[W000 2020-06-06T23:35:51.400-04:00] Primary database cannot be reached.

[W000 2020-06-06T23:35:51.400-04:00] Fast-Start Failover threshold has expired.

[W000 2020-06-06T23:35:51.400-04:00] Try to connect to the standby.

[W000 2020-06-06T23:35:51.400-04:00] Making a last connection attempt to primary database before proceeding with Fast-Start Failover.

[W000 2020-06-06T23:35:51.400-04:00] Check if the standby is ready for failover.

[W000 2020-06-06T23:35:51.405-04:00] **A fast-start failover would have been initiated...**

[W000 2020-06-06T23:35:51.405-04:00] Unable to failover since this observer is in observe-only mode

...

**Note:** The observe-only mode is useful to run the Fast-Start Failover in dry-run mode, where you want to be sure that your infrastructure is configured properly, without false alerts, before having it fully automated.

1. Return to the terminal session on localhost. Using SQL\*Plus, connect as the SYS user and start the primary database. Exit SQL\*Plus.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sat Jun 6 23:50:11 2020

Version 19.3.0.0.0

(c) 1982, 2019, Oracle. All rights reserved. Connected to an idle instance.

SQL> **startup**

ORACLE instance started.

Total System Global Area 629145352 bytes Fixed Size 9137928 bytes Variable Size 377487360 bytes Database Buffers 234881024 bytes

Redo Buffers 7639040 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 [oracle@localhost ~]$

1. Launch the DGMGRL utility and connect as the SYSDG user.

[oracle@localhost ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Sat Jun 6 23:53:33 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg/<password>@orclcdb** Connected to "orclcdb"

Connected as SYSDG. DGMGRL>

1. Stop observer and disable Fast-Start Failover in Observe-only mode.

DGMGRL> **stop observer**

Observer stopped.

DGMGRL> **DISABLE FAST\_START FAILOVER**

Disabled.

DGMGRL>

1. Exit DGMGRL and SQL\*Plus. Leave the terminal windows open for future practices.

## Practice 16-2: Enabling Fast-Start Failover

### Overview

In this practice, you will enable fast-start failover on localhost where the stndby is currently running. After enabling fast-start failover, you will start the observer process.

### Tasks

1. Use a terminal window on localhost connected as oracle with the environment variables set to stndby. Launch the DGMGRL utility and connect as the SYSDG user with operating system authentication.

[oracle@host02 ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@host02 ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Sun Jun 7 00:02:52 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg/<password>@orclcdb** Connected to "orclcdb"

Connected as SYSDG.

DGMGRL>Connected as SYSDG.

1. Verify that there is no Transport Lag or Apply lag at the physical standby database and logical standby database before proceeding with labs.

DGMGRL> **show database orclcdb**

Database - orclcdb

Enterprise Manager Name: orclcdb

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

Real Time Query: ON Instance(s):

orclcdb

Database Status:

SUCCESS

DGMGRL> **show database stndby**

Database - stndby2

Enterprise Manager Name: stndby2.example.com Role: LOGICAL STANDBY

Intended State: APPLY-ON

Transport Lag: *0 seconds* (computed 0 seconds ago) Apply Lag: *0 seconds* (computed 0 seconds ago) Apply Rate: 48.04 MByte/s

Instance(s): stndby2

Database Status:

SUCCESS

1. Display the current configuration and note the current state of fast-start failover.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxAvailability Members:

stndby - Primary database instance

orclcdb - Physical standby database stndby2 - Logical standby database

Members Not Receiving Redo: stndbyFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:

SUCCESS (status updated 61 seconds ago)

DGMGRL>

1. Enable fast-start failover.

DGMGRL> **enable fast\_start failover**

Enabled in Zero Data Loss Mode.

1. Display a detailed status of the current fast-start failover settings.

DGMGRL> **show fast\_start failover**

Fast-Start Failover: Enabled in Zero Data Loss Mode

Protection Mode:

Lag Limit:

MaxAvailability

0 seconds

Threshold: 30 seconds

Active Target: stndby Potential Targets: "stndby"

stndby valid

Observer:

Shutdown Primary:

Auto-reinstate:

(none) TRUE

TRUE

Observer Reconnect: 120 seconds Observer Override: FALSE

Configurable Failover Conditions Health Conditions:

Corrupted Controlfile YES

Corrupted Dictionary YES

Inaccessible Logfile NO

Stuck Archiver NO

Datafile Write Errors YES

Oracle Error Conditions: (none)

DGMGRL>

1. Start the observer process.

DGMGRL> **start observer**

...

**Note:** The prompt will not return after starting the observer. Keep this terminal window open with the observer running in it.

## Practice 16-3: Testing Fast-Start Failover

### Overview

In this practice, you will simulate a disaster on the primary database and observe the automatic failover to the standby database.

### Tasks

1. Use a terminal window on localhost connected as oracle with the environment variables set to orclcdb. Connect to the primary database using SQL\*Plus and simulate a failure by issuing the shutdown abort command. Exit SQL\*Plus.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sun Jun 7 00:10:38 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **shutdown abort** ORACLE instance shut down. SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@localhost ~]$

1. Observe the status output in the terminal session connected to 8c793fb03eed running the observer process from the previous lab step. It may take a minute before the failover is initiated.

DGMGRL> **start observer**

...

2020-06-11T16:23:56.156+00:00

Initiating Fast-Start Failover to database "stndby"...

[S002 2020-06-11T16:23:56.156+00:00] Initiating Fast-start Failover.

Performing failover NOW, please wait...

1. Use a terminal window on stndby connected as oracle with the environment variables set to stndby. Launch the DGMGRL utility and connect as the SYSDG user.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base has been changed from /home/oracle to

/u01/app/oracle [oracle@stndby ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Sun Jun 7 01:14:48 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg/<password>@stndby** Connected to "STNDBY"

Connected as SYSDG.

1. Display the current configuration and note the current state of fast-start failover.

**Note:** A series of error messages such as ORA-16844, ORA-16856, and ORA-16824 might be displayed for a while. Wait until you see only the ORA-16661 message for the orclcdb database.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxAvailability Members:

stndby - Primary database

stndbyFS - Far sync instance

orclcdb - Physical standby database (disabled)

ORA-16661: the standby database needs to be reinstated stndby2 - Logical standby database

Members Not Receiving Redo:

orclcdbFS - Far sync instance

Fast-Start Failover: Enabled in Zero Data Loss Mode Configuration Status:

SUCCESS (status updated 49 seconds ago)

1. Use a terminal window on localhost connected as oracle with the environment variables set to orclcdb. Connect to the former primary database (the new physical standby database) using SQL\*Plus and mount the orclcdb database to initiate reinstatement. Exit SQL\*Plus.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sun Jun 7 01:17:07 2020

Version 19.3.0.0.0

(c) 1982, 2019, Oracle. All rights reserved. Connected to an idle instance.

SQL> **startup mount**

ORACLE instance started.

Total System Global Area 629145352 bytes

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@localhost ~]$

|  |  |  |
| --- | --- | --- |
| Fixed Size | 9137928 | bytes |
| Variable Size | 377487360 | bytes |
| Database Buffers | 234881024 | bytes |
| Redo Buffers | 7639040 | bytes |
| Database mounted.  SQL> **exit** |  |  |

1. Observe the output in the terminal session on 8c793fb03eed running the observer process.

2020-06-07T01:18:53.809-04:00

Initiating reinstatement for database "orclcdb"... Reinstating database "orclcdb", please wait...

[W000 2020-06-07T01:19:15.199-04:00] Primary database cannot be reached.

[W000 2020-06-07T01:19:15.199-04:00] Fast-Start Failover target switch is pending.

[W000 2020-06-07T01:20:09.156-04:00] Primary database cannot be reached.

[W000 2020-06-07T01:20:09.156-04:00] Fast-Start Failover target switch is pending.

[W000 2020-06-07T01:20:49.175-04:00] Primary database cannot be reached.

[W000 2020-06-07T01:20:49.175-04:00] Fast-Start Failover target switch is pending.

Reinstatement of database "orclcdb" succeeded 2020-06-07T01:21:09.990-04:00

[W000 2020-06-07T01:21:09.993-04:00] Successfully reinstated database orclcdb.

[W000 2020-06-07T01:21:15.094-04:00] Disconnecting from database stndby.

1. Return to the DGMGRL session running on stndby and display the configuration.

DGMGRL> **show configuration**

Configuration - DRSolution Protection Mode: MaxAvailability Databases:

stndby - Primary database stndbyFS - Far Sync

orclcdb - (\*) Physical standby database

Warning: ORA-16857: standby disconnected from redo source for longer than specified threshold

stndby2 - Logical standby database orclcdbFS - Far Sync (inactive)

Fast-Start Failover: ENABLED in Zero Data Loss Mode Configuration Status:

WARNING

**Note:** The ORA-\* warning messages depend on the lag and overall performance of the environment. You may or may not see this warning statement. It is a matter of timing.

1. It may take a few moments for the lag for the standby database to clear. Keep displaying the status until it has cleared. Do not continue with labs until the apply lag and transport lag have cleared.

DGMGRL> **show database orclcdb**

Database - orclcdb

Role: PHYSICAL STANDBY

Intended State: APPLY-ON

Transport Lag: 0 seconds (computed 1 second ago) Apply Lag: 0 seconds (computed 1 second ago) Apply Rate: 0 Byte/s

Real Time Query: OFF Instance(s):

orclcdb

Database Status:

SUCCESS

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxAvailability Members:

stndby - Primary database stndbyFS - Far sync instance

orclcdb - (\*) Physical standby database stndby2 - Logical standby database

Members Not Receiving Redo:

orclcdbFS - Far sync instance

Fast-Start Failover: Enabled in Zero Data Loss Mode Configuration Status:

SUCCESS (status updated 55 seconds ago)

DGMGRL>

## Practice 16-4: Switchover to Reinstated Database

### Overview

In this practice, you will perform a switchover to return the configuration to the state that it was before the failover.

### Tasks

1. Validate that the stndby primary database is ready for switchover.

DGMGRL> **validate database stndby**

Database Role:

Primary database

Ready for Switchover: **Yes**

Managed by Clusterware: stndby: NO

Validating static connect identifier for the primary database stndby...

The static connect identifier allows for a connection to database "stndby".

DGMGRL>

1. Validate that the orclcdb standby database is ready for switchover.

DGMGRL> **validate database orclcdb**

Database Role: Physical standby database Primary Database: stndby

Ready for Switchover: **Yes**

Ready for Failover: Yes (Primary Running)

Managed by Clusterware: stndby: NO

orclcdb: NO

Validating static connect identifier for the primary database stndby...

The static connect identifier allows for a connection to database "stndby".

Log Files Cleared:

stndby Standby Redo Log Files: Cleared

orclcdb Online Redo Log Files: Not Cleared orclcdb Standby Redo Log Files: Available

Current Log File Groups Configuration:

Thread # Status

Online Redo Log Groups

Standby Redo Log Groups

(stndby)

1

3

2

Insufficient SRLs

Future Log File Groups Configuration:

Thread # Status

Online Redo Log Groups

Standby Redo Log Groups

(stndby)

1 3 3

Insufficient SRLs

DGMGRL>

1. Switch over to the orclcdb database.

DGMGRL> **switchover to orclcdb**

Performing switchover NOW, please wait...

Operation requires a connection to database "orclcdb" Connecting ...

Connected to "orclcdb" Connected as SYSDG.

New primary database "orclcdb" is opening...

Operation requires start up of instance "stndby" on database "stndby"

Starting instance "stndby"... Connected to an idle instance. ORACLE instance started.

Connected to "stndby" Database mounted.

Database opened. Connected to "stndby" Connected to "orclcdb"

Switchover succeeded, new primary is "orclcdb"

DGMGRL>

1. Display the resulting configuration.

**Note:** A series of error messages such as ORA-16810 and ORA-16786 might be displayed for a while. Wait until the status of the configuration becomes normal.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxAvailability Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby - (\*) Physical standby database stndby2 - Logical standby database

Members Not Receiving Redo:

stndbyFS - Far sync instance

Fast-Start Failover: Enabled in Zero Data Loss Mode Configuration Status:

SUCCESS (status updated 44 seconds ago)

DGMGRL>

1. Stop the observer process.

DGMGRL> **stop observer**

Observer stopped. DGMGRL>

**Note:** The prompt should now be returned in the terminal window that was connected to

host02 running the observer process.

1. Disable fast-start failover.

DGMGRL> **disable fast\_start failover**

Disabled.

1. Reset the protection mode back to maximum performance and exit DGMGRL.

DGMGRL> **edit configuration set protection mode as maxperformance;**

Succeeded.

DGMGRL> **exit**

[oracle@stndby ~]$

# Practices for Lesson 17: Backup and Recovery Considerations in an Oracle Data Guard Configuration

## Practices for Lesson 17: Overview

### Practices Overview

In these practices, you will enable the change tracing feature and setup and configure the recovery manager (RMAN) catalog repository database and use it to perform backup and recovery in a Data Guard environment.

## Practice 17-1: Enable Change Tracking on the Physical Standby Database

### Overview

In this practice, you will enable the change tracking feature on stndby for the physical standby database and verify its usage.

### Tasks

1. Use a terminal window logged in as oracle to stndby with the environment variables set for stndby appropriately. Using operating system authentication, SQL\*Plus connects, by default, to the root container in the lab environment. Enable change tracking by using the file /u01/app/oracle/oradata/stndby/rman\_change\_track.file.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base has been set to /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sun Jun 7 08:48:36 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter database enable block change tracking using file '/u01/app/oracle/oradata/STNDBY/rman\_change\_track.file';**

Database altered. SQL>

1. Verify that block change tracking is enabled, displaying the file name used and file size of the block change tracking file.

SQL> **select filename, status, bytes from v$block\_change\_tracking;**

FILENAME

STATUS

BYTES

/u01/app/oracle/oradata/stndby/rman\_change\_track.file

ENABLED 11599872

1. Exit SQL\*Plus on stndby of the physical standby database. It is recommended to keep the terminal session open with the environment variables set appropriately.

## Practice 17-2: Creating a Recovery Manager Catalog

### Overview

In this practice, you will set up and configure the recovery manager (RMAN) catalog repository database.

### Tasks

1. Open a terminal window connected to localhost as the oracle OS user. Enter <password>

when you are prompted for the password. Set the environment variables to ORCL.

[oracle8c793fb03eed ~]$ **. oraenv** ORACLE\_SID = [ORCL] ? **ORCL**

The Oracle base remains unchanged with value /u01/app/oracle

1. Invoke SQL\*Plus and connect as the SYS user with the SYSDBA privilege. List the PDBs in the connected database.

[oracle8c793fb03eed ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sun Jun 7 12:51:44 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **show pdbs**

CON\_ID CON\_NAME

OPEN MODE RESTRICTED

1. PDB$SEED
2. PDB1
3. EMCCPDB

READ ONLY NO

READ WRITE NO

READ WRITE NO

1. Connect to the RCATPDB PDB.

SQL> **alter session set container=ORCL;**

Session altered. SQL>

1. Determine where the current data files are stored at:

SQL> **select file\_name from dba\_data\_files;**

FILE\_NAME

/u01/app/oracle/oradata/ORCL/pdb1/system01.dbf

/u01/app/oracle/oradata/ORCL/pdb1/sysaux01.dbf

/u01/app/oracle/oradata/ORCL/pdb1/undotbs01.dbf

/u01/app/oracle/oradata/ORCL/pdb1/users01.dbf

SQL>

1. Create a new tablespace for the recovery manager repository using the same storage architecture as the existing files. Name the tablesace rcts and give it an initial size of 30MB with autoextend turned on.

SQL> **create tablespace rcts datafile '/u01/app/oracle/oradata/ORCL/rcts01.dbf' size 30M autoextend on;**

Tablespace created.

1. Create a new schema rcowner setting the default tablespace to the tablespace just created.

SQL> **create user rcowner identified by <password> default tablespace rcts quota unlimited on rcts;**

User created.

1. Grant the recovery catalog owner role to the user just created. Exit SQL\*Plus when done.

SQL> **grant recovery\_catalog\_owner to rcowner;**

Grant succeeded.

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle8c793fb03eed ~]$

1. Use a terminal window on localhost connected as oracle with the environment variables set to orclcdb. Launch the RMAN utility and connect to the rcatpdb service using the account just created.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle

[oracle@localhost ~]$ **rman catalog rcowner/<password>@em13c:1521/ORCL**

Recovery Manager: Release 19.0.0.0.0 - Production on Sun Jun 7 13:05:44 2020

Version 19.3.0.0.0

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

connected to recovery catalog database

1. Create the recovery catalog.

RMAN> **create catalog;**

recovery catalog created

## Practice 17-3: Registering Your Database in the Recovery Catalog

### Overview

In this practice, you register the primary database in the recovery catalog.

### Tasks

1. Return to the RMAN session on localhost. Connect to the orclcdb primary database with

SYSDBA privilege and register the database.

RMAN> **connect target 'sys/<password>@orclcdb as sysdba'** connected to target database: ORCLCDB (DBID=2732402101) RMAN> **register database;**

database registered in recovery catalog starting full resync of recovery catalog full resync complete

RMAN>

**Note:** Your DBID may be different.

1. List the DB\_UNIQUE\_NAME for all databases known to the recovery catalog.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| RMAN> **list db\_unique\_name of database;** | | | | | | | |
| List of DB Key |  | Databases DB Name | DB ID |  | Database Role |  | Db\_unique\_name |
| 1 ORCLCDB 2732402101 PRIMARY ORCLCDB | | | | | | | |

1. Generate a schema report for the orclcdb primary database.

RMAN> **report schema for db\_unique\_name orclcdb;**

Report of database schema for database with db\_unique\_name ORCLCDB

List of Permanent Datafiles

===========================

File Size(MB) Tablespace

RB segs Datafile Name

---

1 960 SYSTEM YES

/u01/app/oracle/oradata/ORCLCDB/system01.dbf

1. 940 SYSAUX NO

/u01/app/oracle/oradata/ORCLCDB/sysaux01.dbf

1. 280 UNDOTBS1 YES

/u01/app/oracle/oradata/ORCLCDB/undotbs01.dbf

1. 270 PDB$SEED:SYSTEM NO

/u01/app/oracle/oradata/ORCLCDB/pdbseed/system01.dbf

1. 330 PDB$SEED:SYSAUX NO

/u01/app/oracle/oradata/ORCLCDB/pdbseed/sysaux01.dbf

1. 5 USERS NO

/u01/app/oracle/oradata/ORCLCDB/users01.dbf

1. 100 PDB$SEED:UNDOTBS1 NO

/u01/app/oracle/oradata/ORCLCDB/pdbseed/undotbs01.dbf

1. 510 DEV1:SYSTEM YES

/u01/app/oracle/oradata/ORCLCDB/dev1/system01.dbf

1. 370 DEV1:SYSAUX NO

/u01/app/oracle/oradata/ORCLCDB/dev1/sysaux01.dbf

1. 100 DEV1:UNDOTBS1 YES

/u01/app/oracle/oradata/ORCLCDB/dev1/undotbs01.dbf

1. 5 DEV1:USERS NO

/u01/app/oracle/oradata/ORCLCDB/dev1/users01.dbf

List of Temporary Files

=======================

File Size(MB) Tablespace Maxsize(MB) Tempfile Name

---

1 132 TEMP 32767

/u01/app/oracle/oradata/ORCLCDB/temp01.dbf

2 36 PDB$SEED:TEMP 32767

/u01/app/oracle/oradata/ORCLCDB/pdbseed/temp012020-06-04\_02-09- 11-127-AM.dbf

3 36 DEV1:TEMP 32767

/u01/app/oracle/oradata/ORCLCDB/dev1/temp01.dbf

RMAN>

1. List all the archive logs for the orclcdb primary database.

RMAN> **list archivelog all for db\_unique\_name orclcdb;**

List of Archived Log Copies for database with db\_unique\_name ORCLCDB

================================================================

=====

Key

Thrd Seq

S Low Time

-

191

1

Name:

6

A 04-JUN-20

/u01/app/oracle/fast\_recovery\_area/ORCLCDB/archivelog/2020\_06\_04/ o1\_mf\_1\_6\_hfjpp4gy\_.arc

192

1

Name:

7

A 04-JUN-20

/u01/app/oracle/fast\_recovery\_area/ORCLCDB/archivelog/2020\_06\_04/ o1\_mf\_1\_7\_hfk3hmrg\_.arc

...

328

1

Name:

16

A 11-JUN-20

/u01/app/oracle/fast\_recovery\_area/ORCLCDB/archivelog/2020\_06\_11/ o1\_mf\_1\_16\_hg4qd0d3\_.arc

RMAN>

1. Display all the current configuration parameters for the orclcdb primary database.

#### RMAN> show all for db\_unique\_name orclcdb;

RMAN configuration parameters for database with db\_unique\_name ORCLCDB are:

CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default CONFIGURE BACKUP OPTIMIZATION OFF; # default CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default CONFIGURE CONTROLFILE AUTOBACKUP ON; # default

CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO

'%F'; # default

CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO

BACKUPSET; # default

CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; #

default

CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; #

default

CONFIGURE MAXSETSIZE TO UNLIMITED; # default CONFIGURE ENCRYPTION FOR DATABASE OFF; # default CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default

CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR LOAD TRUE ; # default

CONFIGURE RMAN OUTPUT TO KEEP FOR 7 DAYS; # default CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default

CONFIGURE SNAPSHOT CONTROLFILE NAME TO

'/u01/app/oracle/product/19.3.0/dbhome\_1/dbs/snapcf\_orclcdb.f'; # default

RMAN>

## Practice 17-4: Configuring RMAN Parameters

### Overview

In this practice, you will configure RMAN for use in a Data Guard environment.

### Tasks

1. In your RMAN session (connected to your primary database), configure the backup retention policy to allow for recovery for seven days.

RMAN> **configure retention policy to recovery window of 7 days;**

new RMAN configuration parameters:

CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;

new RMAN configuration parameters are successfully stored starting full resync of recovery catalog

full resync complete

1. Specify that archived redo log files can be deleted after they are applied to the standby database.

RMAN> **configure archivelog deletion policy to applied on all standby;**

new RMAN configuration parameters:

CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON ALL STANDBY;

new RMAN configuration parameters are successfully stored starting full resync of recovery catalog

full resync complete

1. Configure the connect identifier for your primary database.

RMAN> **configure db\_unique\_name orclcdb connect identifier 'orclcdb';**

new RMAN configuration parameters:

CONFIGURE DB\_UNIQUE\_NAME 'orclcdb' CONNECT IDENTIFIER 'orclcdb';

new RMAN configuration parameters are successfully stored starting full resync of recovery catalog

full resync complete

1. Configure the connect identifier for your physical standby database.

RMAN> **configure db\_unique\_name stndby connect identifier 'stndby';**

new RMAN configuration parameters:

CONFIGURE DB\_UNIQUE\_NAME 'stndby' CONNECT IDENTIFIER 'stndby';

new RMAN configuration parameters are successfully stored starting full resync of recovery catalog

full resync complete

1. Your physical standby database is registered with the recovery catalog. Use the LIST DB\_UNIQUE\_NAME command to see the registration information about your primary and standby databases.

RMAN> **list db\_unique\_name of database;**

List of Databases

DB Key DB Name DB ID Database Role

Db\_unique\_name

1

1

ORCLCDB 2732402101 PRIMARY

ORCLCDB 2732402101 STANDBY

ORCLCDB

STNDBY

1. Use the REPORT SCHEMA command to view additional information about your physical standby database. Exit RMAN when done.

RMAN> **report schema for db\_unique\_name stndby;**

Report of database schema for database with db\_unique\_name STNDBY

List of Permanent Datafiles

===========================

RMAN> **exit;**

Recovery Manager complete. [oracle@localhost ~]$

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| File |  | Size(MB) |  | Tablespace |  | RB segs |  | Datafile Name |
| --- |  |  |  |  |  |  |  |  |
| 1 |  | 960 |  | SYSTEM |  | YES |  |  |
| 3 |  | 940 |  | SYSAUX |  | NO |  |  |
| 4 |  | 280 |  | UNDOTBS1 |  | YES |  |  |
| 5 |  | 270 |  | PDB$SEED:SYSTEM |  | NO |  |  |
| 6 |  | 330 |  | PDB$SEED:SYSAUX |  | NO |  |  |
| 7 |  | 5 |  | USERS |  | NO |  |  |
| 8 |  | 100 |  | PDB$SEED:UNDOTBS1 |  | NO |  |  |
| 9 |  | 510 |  | DEV1:SYSTEM |  | YES |  |  |
| 10 |  | 370 |  | DEV1:SYSAUX |  | NO |  |  |
| 11 |  | 100 |  | DEV1:UNDOTBS1 |  | YES |  |  |
| 12 |  | 5 |  | DEV1:USERS |  | NO |  |  |

## Practice 17-5: Recovering a Data File on Your Primary Database Over the Network

### Overview

In this practice, you recover a data file in your primary database by using a data file from your physical standby database. You will create a new data file in order to simulate a disaster, without affecting the existing data files on the primary database.

### Tasks

1. The logical standby database does not honor the DB\_FILE\_NAME\_CONVERT parameter. This will cause an error when a tablespace is created on the primary database and force the Logical Apply process to shut down because the directory doesn't exit. Use a terminal window logged in as oracle to stndby. Create a symbolic link ORCLCDB linking to stndby2 so that file creation can proceed.

[oracle@stndby ~]$ **cd /u01/app/oracle/oradata**

[oracle@stndby oradata]$ **ln -s stndby2 ORCLCDB**

**Note:** The Data Guard documentation shows how to create a DDL handler using a procedure, along with the built-in DBMS\_LOGSTDBY.SKIP procedure to skip over the DDL with the wrong path names and invoke the handler to rename the path in the command.

1. Use a terminal window logged in as oracle to localhost with the environment variables set for orclcdb appropriately. Launch SQL\*Plus and create a new tablespace SAMPLE in the DEV1 pluggable database with a data file

/u01/app/oracle/oradata/orclcdb/dev1/sample01.dbf and a size of 5 MB.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sun Jun 7 09:19:53 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter session set container=DEV1;**

Session altered.

SQL> **create tablespace SAMPLE datafile '/u01/app/oracle/oradata/ORCLCDB/dev1/sample01.dbf' size 5M;**

Tablespace created. SQL>

1. Connect to your physical standby instance as the SYSDBA user and show the

standby\_file\_management parameter.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SQL> **connect sys/<password>@stndby as sysdba**  Connected.  SQL> **show parameter standby\_file\_management** | | | | |
| NAME |  | TYPE |  | VALUE |
| standby\_file\_management SQL> |  | string |  | AUTO |

1. Verify that the data file has successfully been created on the physical standby database.

SQL> **col name format a65**

SQL> **select file#,name from v$datafile;**

FILE# NAME

1 /u01/app/oracle/oradata/STNDBY/system01.dbf

1. /u01/app/oracle/oradata/STNDBY/sysaux01.dbf
2. /u01/app/oracle/oradata/STNDBY/undotbs01.dbf
3. /u01/app/oracle/oradata/STNDBY/pdbseed/system01.dbf
4. /u01/app/oracle/oradata/STNDBY/pdbseed/sysaux01.dbf
5. /u01/app/oracle/oradata/STNDBY/users01.dbf
6. /u01/app/oracle/oradata/STNDBY/pdbseed/undotbs01.dbf
7. /u01/app/oracle/oradata/STNDBY/dev1/system01.dbf
8. /u01/app/oracle/oradata/STNDBY/dev1/sysaux01.dbf
9. /u01/app/oracle/oradata/STNDBY/dev1/undotbs01.dbf
10. /u01/app/oracle/oradata/STNDBY/dev1/users01.dbf

FILE# NAME

25 /u01/app/oracle/oradata/STNDBY/dev1/sample01.dbf

12 rows selected.

SQL>

1. Connect to your logical standby instance as the SYSDBA user.

SQL> **connect sys/<password>@stndby2 as sysdba**

Connected.

1. Verify that the data file has successfully been created on the logical standby database.

#### SQL> select file#,name from v$datafile;

FILE# NAME

1 /u01/app/oracle/oradata/stndby2/system01.dbf

1. /u01/app/oracle/oradata/stndby2/sysaux01.dbf
2. /u01/app/oracle/oradata/stndby2/undotbs01.dbf
3. /u01/app/oracle/oradata/stndby2/pdbseed/system01.dbf
4. /u01/app/oracle/oradata/stndby2/pdbseed/sysaux01.dbf
5. /u01/app/oracle/oradata/stndby2/users01.dbf
6. /u01/app/oracle/oradata/stndby2/pdbseed/undotbs01.dbf
7. /u01/app/oracle/oradata/stndby2/dev1/system01.dbf
8. /u01/app/oracle/oradata/stndby2/dev1/sysaux01.dbf
9. /u01/app/oracle/oradata/stndby2/dev1/undotbs01.dbf
10. /u01/app/oracle/oradata/stndby2/dev1/users01.dbf

FILE# NAME

25 /u01/app/oracle/oradata/ORCLCDB/dev1/sample01.dbf

12 rows selected.

SQL>

1. Reconnect to your primary database and create the hr.employees2 table as a copy of the hr.employees table into the newly created tablespace. Exit SQL\*Plus.

SQL> **connect system/<password>@localhost:1521/DEV1.example.com**

Connected.

SQL> **create table hr.employees2 tablespace sample as select \* from hr.employees;**

Table created.

1. Verify that the table was created by counting the number of rows it contains.

SQL> **select count(\*) from hr.employees2;**

COUNT(\*)

107

1. Move the sample01.dbf file to sample01.sav to simulate a failure in the primary database.

SQL> **!mv /u01/app/oracle/oradata/ORCLCDB/dev1/sample01.dbf**

**/u01/app/oracle/oradata/ORCLCDB/dev1/sample01.sav**

1. Connect to the root container and shut down abort the primary database. Exit SQL\*Plus.

SQL> **connect sys/<password>@orclcdb as sysdba**

Connected.

SQL> **shutdown abort**

ORACLE instance shut down.

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@localhost ~]$

1. Launch SQL\*Plus and start the database instance. Exit SQL\*Plus.

[oracle@localhost]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sun Jun 7 09:29:05 2020

Version 19.3.0.0.0

(c) 1982, 2019, Oracle. All rights reserved. Connected to an idle instance.

|  |  |  |
| --- | --- | --- |
| Fixed Size | 9137928 | bytes |
| Variable Size | 377487360 | bytes |
| Database Buffers | 234881024 | bytes |
| Redo Buffers | 7639040 | bytes |
| Database mounted. |  |  |
| Database opened. |  |  |

1. Use RMAN to restore the missing datafile using the physical standby database over the network. Exit RMAN when done.

[oracle@localhost ~]$ **rman target sys/<password>@orclcdb**

Recovery Manager: Release 19.0.0.0.0 - Production on Sun Jun 7 09:37:22 2020

Version 19.3.0.0.0

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

connected to target database: ORCLCDB (DBID=2732402101) RMAN> **restore datafile 25 from service 'stndby';**

Starting restore at 07-JUN-20

using target database control file instead of recovery catalog

SQL> **startup**

ORACLE instance started.

Total System Global Area 629145352 bytes

SQL> **alter pluggable database dev1 open;**

alter pluggable database dev1 open

\*

ERROR at line 1:

ORA-01157: cannot identify/lock data file 25 - see DBWR trace file

ORA-01110: data file 25: '/u01/app/oracle/oradata/ORCLCDB/dev1/sample01.dbf'

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@localhost ~]$

allocated channel: ORA\_DISK\_1

channel ORA\_DISK\_1: SID=43 device type=DISK

channel ORA\_DISK\_1: starting datafile backup set restore

channel ORA\_DISK\_1: using network backup set from service stndby

channel ORA\_DISK\_1: specifying datafile(s) to restore from backup set

channel ORA\_DISK\_1: restoring datafile 00025 to

/u01/app/oracle/oradata/ORCLCDB/dev1/sample01.dbf

channel ORA\_DISK\_1: restore complete, elapsed time: 00:00:02 Finished restore at 07-JUN-20

RMAN> **recover datafile 25;**

Starting recover at 07-JUN-20 using channel ORA\_DISK\_1

starting media recovery

media recovery complete, elapsed time: 00:00:00 Finished recover at 07-JUN-20

RMAN> **exit**

Recovery Manager complete. [oracle@localhost ~]$

1. Launch SQL\*Plus and switch the container to the pluggable database.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sun Jun 7 09:41:20 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **show pdbs**

1. Verify that the table has been recovered and the rows exist in the table.

SQL> **alter session set container = DEV1;**

Session altered.

SQL> **select \* from hr.employees2;**

...

EMPLOYEE\_ID FIRST\_NAME LAST\_NAME

EMAIL

SALARY

PHONE\_NUMBER

HIRE\_DATE JOB\_ID

COMMISSION\_PCT MANAGER\_ID DEPARTMENT\_ID

205 Shelley

SHIGGINS

12008

Higgins

515.123.8080

07-JUN-02 AC\_MGR

101 110

206 William

WGIETZ

8300

Gietz

515.123.8181

07-JUN-02 AC\_ACCOUNT

205 110

EMPLOYEE\_ID FIRST\_NAME

LAST\_NAME

EMAIL

SALARY

PHONE\_NUMBER

HIRE\_DATE JOB\_ID

COMMISSION\_PCT MANAGER\_ID DEPARTMENT\_ID

CON\_ID CON\_NAME

OPEN MODE RESTRICTED

1. PDB$SEED
2. DEV1

READ ONLY NO

MOUNTED

SQL>

SQL> **alter pluggable database dev1 open;**

Pluggable database altered.

107 rows selected.

1. Drop the tablespace that was created in the lab along with the data files. Exit SQL\*Plus.

SQL> **drop tablespace sample including contents and datafiles;**

Tablespace dropped; SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@localhost ~]$

## Practice 17-6: Rolling Forward a Standby Database with One Command

### Overview

In this practice, you will resolve problems such as missing or corrupted archive log file, an unrecoverable archive gap, or the need to roll standby forward in time without applying a large number of archivelog files.

### Tasks

1. Use the terminal window on localhost as the oracle user. Make sure that you set up your environment variables correctly.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [orclcdb] ? **orclcdb**

The Oracle base remains unchanged with value /u01/app/oracle [oracle@localhost ~]$

1. Disable the redo transport service in preparation of the practice.

[oracle@localhost ~]$ **dgmgrl**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Sun Jun 7 10:35:59 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. DGMGRL> **connect sysdg/<password>@orclcdb** Connected to "orclcdb"

Connected as SYSDG.

DGMGRL> **edit database orclcdb set state='TRANSPORT-OFF';**

Succeeded.

DGMGRL>

1. Use the terminal window on stndby as the oracle user. Make sure that you set up your environment variables correctly. Launch SQL\*Plus to stop the stndby standby database.

[oracle@stndby ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **stndby**

The Oracle base has been set to /u01/app/oracle [oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sun Jun 7 10:39:29 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **shutdown immediate**

Database closed. Database dismounted.

ORACLE instance shut down. SQL>

1. Open a new terminal window on localhost. Then make a note of the current log sequence number. Record the sequence number of the online redo log file in thread 1. In your case, the sequence# is 82)

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sun Jun 7 10:41:45 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **select thread#, sequence# from v$log where status='CURRENT' order by 1;**

THREAD# SEQUENCE#

1

82

SQL>

1. In the DEV1 PDB, create a simple table named hr.test17 and insert a few rows.

SQL> **alter session set container=DEV1;**

Session altered.

SQL> **@/home/oracle/setup/setup\_17-6.sql**

SQL> create table hr.test17 (col number); Table created.

SQL> insert into hr.test17 values (1);

1 row created.

SQL> insert into hr.test17 values (1);

1 row created.

SQL> insert into hr.test17 values (1);

1 row created.

SQL> commit;

Commit complete.

SQL>

1. Switch the current log file to advance the online redo log sequence number.

SQL> **connect / as sysdba**

Connected.

SQL> **alter system switch logfile;**

System altered.

SQL> **SELECT THREAD#, MAX(SEQUENCE#) FROM V$ARCHIVED\_LOG WHERE RESETLOGS\_CHANGE# = (SELECT MAX(RESETLOGS\_CHANGE#) FROM V$ARCHIVED\_LOG) GROUP BY THREAD#;**

2 3

THREAD# MAX(SEQUENCE#)

1

82

SQL>

1. Identify the most current archived log files by using the number identified in step 6.

**Note:** If there are more than one entries, choose the latest archived log file. Exit SQL\*Plus.

SQL> **col name format a65**

SQL> **select thread#, name from v$archived\_log where thread#=1 and sequence#=82;**

THREAD#

NAME

1

orclcdbfs

1

/u01/app/oracle/fast\_recovery\_area/ORCLCDB/archivelog/**2020\_06\_05**/o 1\_mf\_1\_82\_hfnrhsbz\_.arc

1

/u01/app/oracle/fast\_recovery\_area/ORCLCDB/archivelog/**2020\_06\_07**/o 1\_mf\_1\_82\_hft449l6\_.arc

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0 [oracle@localhost ~]$

1. Now, simulate a loss of the archived log file before transferring to the standby database. Remove the archived log file identified in the previous step.

[oracle@localhost ~]$ **rm**

**/u01/app/oracle/fast\_recovery\_area/ORCLCDB/archivelog/2020\_06\_07/o1**

**\_mf\_1\_82\_hft449l6\_.arc**

[oracle@localhost ~]$

1. Return to the DGMGRL session on localhost. Start the redo transport service to the physical standby database.

DGMGRL> **edit database orclcdb set state='TRANSPORT-ON';**

Succeeded.

DGMGRL>

1. Return to the SQL\*Plus session on stndby and start the physical standby database.

SQL> **startup**

ORACLE instance started.

Total System Global Area 629145352 bytes

Fixed Size Variable Size

9137928 bytes

377487360 bytes

Database Buffers 234881024 bytes

Redo Buffers 7639040 bytes Database mounted.

Database opened.

SQL> **alter pluggable database dev1 open;**

Pluggable database altered.

1. In the DEV1 PDB, verify that the physical standby is synchronized with the primary database. Exit SQL\*Plus.

SQL> **alter session set container=DEV1;**

Session altered.

SQL> **select \* from hr.test17;**

select \* from hr.test17

\*

ERROR at line 1:

ORA-00942: table or view does not exist

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0 [oracle@stndby ~]$

**Note:** The changes made to the primary database have not been applied due to the missing archived log files in the primary database. Remember that you have removed the most current archived log files to simulate the unrecoverable archived log file gap issue.

1. Return to the DGMGRL session on localhost, stop the Managed Recovery Process to prepare for the standby database recovery.

DGMGRL> **edit database stndby set state='APPLY-OFF';**

Succeeded.

DGMGRL>

1. Return to the terminal session on stndby. Let’s see how we can refresh the standby database with one command in case of the unresolvable scenario. Launch the RMAN utility to recover the standby database with one command. Exit the RMAN utility.

[oracle@stndby ~]$ **rman target /**

Recovery Manager: Release 19.0.0.0.0 - Production on Sun Jun 7 11:28:41 2020

Version 19.3.0.0.0

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

connected to target database: ORCLCDB (DBID=2732274290, not open) RMAN> **RECOVER STANDBY DATABASE FROM SERVICE=orclcdb;**

Starting recover at 07-JUN-20

using target database control file instead of recovery catalog Executing: alter database flashback off

Executing: alter database disable block change tracking Oracle instance started

Total System Global Area 629145352 bytes

Fixed Size 9137928 bytes Variable Size 373293056 bytes

Database Buffers 239075328 bytes Redo Buffers 7639040 bytes

contents of Memory Script:

{

restore standby controlfile from service 'orclcdb'; alter database mount standby database;

}

executing Memory Script

Starting restore at 07-JUN-20 allocated channel: ORA\_DISK\_1

channel ORA\_DISK\_1: SID=20 device type=DISK

channel ORA\_DISK\_1: starting datafile backup set restore channel ORA\_DISK\_1: using network backup set from service orclcdb channel ORA\_DISK\_1: restoring control file

channel ORA\_DISK\_1: restore complete, elapsed time: 00:00:02 output file name=/u01/app/oracle/oradata/STNDBY/control01.ctl output file name=/u01/app/oracle/fast\_recovery\_area/STNDBY/control02.ctl Finished restore at 07-JUN-20

released channel: ORA\_DISK\_1 Statement processed

Executing: alter system set standby\_file\_management=manual

contents of Memory Script:

{

recover database from service 'orclcdb';

}

executing Memory Script

Starting recover at 07-JUN-20

Starting implicit crosscheck backup at 07-JUN-20 allocated channel: ORA\_DISK\_1

channel ORA\_DISK\_1: SID=24 device type=DISK Crosschecked 14 objects

Finished implicit crosscheck backup at 07-JUN-20

Starting implicit crosscheck copy at 07-JUN-20 using channel ORA\_DISK\_1

Crosschecked 2 objects

Finished implicit crosscheck copy at 07-JUN-20

searching for all files in the recovery area cataloging files...

cataloging done

List of Cataloged Files

=======================

File Name:

/u01/app/oracle/fast\_recovery\_area/STNDBY/archivelog/2020\_06\_07/o1

\_mf\_1\_60\_hfry8lng\_.arc

...

File Name:

/u01/app/oracle/fast\_recovery\_area/STNDBY/autobackup/2020\_06\_06/o1

\_mf\_s\_1042403484\_hfrfpq8w\_.bkp

using channel ORA\_DISK\_1

skipping datafile 5; already restored to SCN 1944601 skipping datafile 6; already restored to SCN 1944601 skipping datafile 8; already restored to SCN 1944601 channel ORA\_DISK\_1: starting incremental datafile backup set restore

channel ORA\_DISK\_1: using network backup set from service orclcdb destination for restore of datafile 00001:

/u01/app/oracle/oradata/STNDBY/system01.dbf

channel ORA\_DISK\_1: restore complete, elapsed time: 00:00:35 channel ORA\_DISK\_1: starting incremental datafile backup set restore

channel ORA\_DISK\_1: using network backup set from service orclcdb destination for restore of datafile 00003:

/u01/app/oracle/oradata/STNDBY/sysaux01.dbf

channel ORA\_DISK\_1: restore complete, elapsed time: 00:00:35 channel ORA\_DISK\_1: starting incremental datafile backup set

...

starting media recovery

archived log for thread 1 with sequence xx is already on disk as file

/u01/app/oracle/fast\_recovery\_area/STNDBY/archivelog/2020\_06\_23/o1

\_mf\_1\_37\_hh53htbg\_.arc archived log file

name=/u01/app/oracle/fast\_recovery\_area/STNDBY/archivelog/2020\_06\_ 23/o1\_mf\_1\_37\_hh53htbg\_.arc thread=1 sequence=37

media recovery complete, elapsed time: 00:00:01 Finished recover at 23-JUN-20

Reenabling controlfile options for auxiliary database

Executing: alter database enable block change tracking using file '/u01/app/oracle/oradata/STNDBY/rman\_change\_track.file' Executing: alter system set standby\_file\_management=auto

Finished recover at 23-JUN-20 RMAN> **exit**

Recovery Manager complete. [oracle@stndby ~]$

1. Using SQL\*Plus, connect to the stndby database. Start the database and its PDB.

[oracle@stndby ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sun Jun 7 11:34:10 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **show pdbs**

CON\_ID CON\_NAME

OPEN MODE RESTRICTED

1. PDB$SEED
2. DEV1

MOUNTED MOUNTED

SQL> **alter database open;**

Database altered.

SQL> **alter pluggable database dev1 open;**

Pluggable database altered.

1. Return to the DGMGRL session on localhost. Start the redo apply service.

DGMGRL> **edit database stndby set state='APPLY-ON';**

Succeeded.

DGMGRL>

1. Return to the SQL\*Plus session on stndby connected to the stndby database. In the DEV1 PDB, verify that the physical standby applies the change made to the primary database. Exit SQL\*Plus.

SQL> **alter session set container=DEV1;**

Session altered.

SQL> **select \* from hr.test17;**

COL 1

1

1

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0 [oracle@stndby ~]$

1. Return to the DGMGRL session on localhost. Display the status of the data guard broker configuration.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby - Physical standby database stndby2 - Logical standby database

Warning: ORA-16809: multiple warnings detected for the

member

Members Not Receiving Redo: stndbyFS - Far sync instance

Fast-Start Failover: Disabled Configuration Status:

WARNING (status updated 49 seconds ago)

DGMGRL>

**Note:** In this practice, we tested how to recover the physical standby database with a single command in case of the unrecoverable redo gap scenario. The ORA-\* warning message in the stndby2 standby database was caused by the missing redo from the primary database.

1. Disable the stndby2 logical standby database.

**Note:** We can safely disable the stndby2 database because it will not be used in later practices.

DGMGRL> **disable database stndby2;**

Disabled.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby - Physical standby database

stndby2 - Logical standby database (disabled) ORA-16749: The member was disabled manually.

Members Not Receiving Redo:

stndbyFS - Far sync instance

Fast-Start Failover: Disabled Configuration Status:

SUCCESS (status updated 48 seconds ago)

DGMGRL>

1. Exit DGMGRL and SQL\*Plus leaving the terminal window open for future practices.

# Practices for Lesson 18: Enhanced Client Connectivity in a Data Guard Environment

## Practices for Lesson 18: Overview

### Practices Overview

In these practices, you will create a service to connect to the DEV1 pluggable database, and also create a database startup trigger that will start the service on any host machine that the primary database is running on. You will also modify the service to support Application Continuity and test it.

## Practice 18-1: Creating and Testing Primary Database Services

### Overview

In this practice, you will create and test a service for the DEV1 pluggable database on the primary database, and follow that service as it migrates from localhost to stndby during switchover exercises.

### Tasks

1. Use a terminal window logged in as oracle to localhost with the environment variables set for orclcdb appropriately. Launch SQL\*Plus and set the session container to the DEV1 pluggable database.

[oracle@localhost ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sun Jun 7 14:44:47 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter session set container=dev1;**

Session altered.

1. Create and start a service with the name PRMY

SQL> **exec DBMS\_SERVICE.CREATE\_SERVICE('PRMY','PRMY')**

PL/SQL procedure successfully completed.

SQL> **exec DBMS\_SERVICE.START\_SERVICE('PRMY')**

PL/SQL procedure successfully completed.

1. From within SQL\*Plus, display the status of the Oracle listener running on localhost and verify that the service was started successfully. Do not exit SQL\*Plus.

SQL> **!lsnrctl status**

LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 20-DEC-2013 16:06:01

(c) 1991, 2013, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost.example.com)(PORT= 1521)(SEND\_SDU=10485760)(RECV\_SDU=10485760)))

STATUS of the LISTENER

Alias LISTENER

Version TNSLSNR for Linux: Version 12.1.0.1.0 - Production

Start Date 18-DEC-2013 10:25:42

Uptime 2 days 5 hr. 40 min. 21 sec

Trace Level off

Security ON: Local OS Authentication

SNMP OFF

Listener Parameter File

/u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/listener.ora

Listener Log File

/u01/app/oracle/diag/tnslsnr/localhost/listener/alert/log.xml Listening Endpoints Summary...

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=localhost.example.com)(PORT= 1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=localhost.example.com)(PORT

=5500))(Security=(my\_wallet\_directory=/u01/app/oracle/admin/orclcdb/ xdb\_wallet))(Presentation=HTTP)(Session=RAW))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=localhost.example.com)(PORT

=5501))(Security=(my\_wallet\_directory=/u01/app/oracle/admin/orclcdb/ xdb\_wallet))(Presentation=HTTP)(Session=RAW))

Services Summary...

***Service "PRMY" has 1 instance(s).***

***Instance "orclcdb", status READY, has 1 handler(s) for this service...***

Service "orclcdb.example.com" has 2 instance(s).

Instance "orclcdb", status UNKNOWN, has 1 handler(s) for this service...

Instance "orclcdb", status READY, has 1 handler(s) for this service...

Service "orclcdbXDB.example.com" has 1 instance(s).

Instance "orclcdb", status READY, has 0 handler(s) for this service...

Service "orclcdb\_DGB.example.com" has 1 instance(s).

Instance "orclcdb", status READY, has 1 handler(s) for this service...

Service "orclcdb\_DGMGRL.example.com" has 1 instance(s).

Instance "orclcdb", status UNKNOWN, has 1 handler(s) for this service...

Service "dev1.example.com" has 1 instance(s).

Instance "orclcdb", status READY, has 1 handler(s) for this service...

The command completed successfully

1. Use another terminal window logged in as oracle to host02 with the environment variables set for orclcdbFS appropriately. Launch SQL\*Plus and connect to the PRMY.EXAMPLE.COM service.

[oracle@host02 ~]$ **. oranev**

ORACLE\_SID = [oracle] ? **orclcdbFS**

The Oracle base has been set to /u01/app/oracle

[oracle@host02 ~]$ **sqlplus system/<password>@prmy**

SQL\*Plus: Release 19.0.0.0.0 - Production on Sun Jun 7 14:47:38 2020

Version 19.3.0.0.0

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

Last Successful login time: Sun Jun 07 2020 09:26:10 -04:00 Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL>

**Note:** The tnsnames.ora network configuration file was already created, with an entry PRMY that attempts to connect to both localhost.example.com and stndby.example.com, and access a service name of PRMY.EXAMPLE.COM.

1. Verify that you are indeed connected to the orclcdb instance (the primary database).

SQL> **select instance\_name from v$instance;**

INSTANCE\_NAME

orclcdb

1. Verify that your connection has been established with the DEV1 pluggable database and not the root container.

SQL> **select sys\_context ('USERENV', 'CON\_NAME') as container FROM dual;**

CONTAINER DEV1

SQL>

1. Return to the SQL\*Plus session running on localhost for the primary database. Create an on database startup trigger in the DEV1 PDB that will open the DEV1 pluggable database if it is not already open. The trigger should then start the PRMY.EXAMPLE.COM service after it switches the container to the DEV1 container. The logic should only execute if the database is in the primary role. Exit SQL\*Plus on localhost.

SQL> **show con\_name**

CON\_NAME DEV1

SQL> **@/home/oracle/setup/create\_trigger.sql**

11

12

13

execute immediate 'alter pluggable database dev1 open'; end if;

dbms\_service.start\_service('PRMY');

|  |  |  |
| --- | --- | --- |
| SQL> | create or replace trigger primary\_services |  |
| 2 | after startup on database |
| 3 | declare |
| 4 | role varchar2(30); |
| 5 | omode varchar2(30); |
| 6 | begin |
| 7 | select database\_role into role from v$database; |
| 8 | select open\_mode into omode from v$pdbs where name | = 'DEV1'; |
| 9 | if role = 'PRIMARY' then |  |
| 10 | if omode != 'READ WRITE' then |  |

1. end if;
2. end;

16 /

Trigger created.

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@localhost ~]

1. Launch DGMGRL and connect to the SYSDG account. Show the configuration.

[oracle@localhost ~]$ **dgmgrl sysdg/<password>@orclcdb**

DGMGRL for Linux: Release 19.0.0.0.0 - Production on Tue Jun 23 20:09:47 2020

Version 19.3.0.0.0

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

Welcome to DGMGRL, type "help" for information. Connected to "orclcdb"

Connected as SYSDG. DGMGRL>

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby - Physical standby database

stndby2 - Logical standby database (disabled) ORA-16749: The member was disabled manually.

Members Not Receiving Redo:

Fast-Start Failover: Disabled

Configuration Status:

SUCCESS (status updated 48 seconds ago)

DGMGRL>

1. Validate that the primary and physical standby databases are ready for switchover.

DGMGRL> **validate database orclcdb**

Database Role:

Primary database

Ready for Switchover: Yes

Flashback Database Status: orclcdb: On

stndby: Off

Managed by Clusterware: orclcdb: NO

Validating static connect identifier for the primary database orclcdb...

The static connect identifier allows for a connection to database "orclcdb".

DGMGRL> **validate database stndby**

Database Role: Physical standby database Primary Database: orclcdb

Ready for Switchover: Yes

Ready for Failover: Yes (Primary Running)

Flashback Database Status: orclcdb: On

stndby: Off

Managed by Clusterware: orclcdb: NO

stndby: NO

Validating static connect identifier for the primary database orclcdb...

The static connect identifier allows for a connection to database "orclcdb".

Current Log File Groups Configuration:

Thread # Status

Online Redo Log Groups

Standby Redo Log Groups

(stndby)

1 3 3

Insufficient SRLs

Future Log File Groups Configuration:

Thread # Status

Online Redo Log Groups

Standby Redo Log Groups

(stndby)

1 3 2

Insufficient SRLs

1. Perform a switch over to the stndby physical standby database. Do not exit DGMGRL.

DGMGRL> **switchover to stndby**

Performing switchover NOW, please wait...

Operation requires a connection to database "stndby" Connecting ...

Connected to "stndby" Connected as SYSDG.

New primary database "stndby" is opening...

Operation requires start up of instance "orclcdb" on database "orclcdb"

Starting instance "orclcdb"... Connected to an idle instance. ORACLE instance started.

Connected to "orclcdb" Database mounted.

Database opened. Connected to "orclcdb"

Switchover succeeded, new primary is "stndby"

DGMGRL>

1. Return to the SQL\*Plus session on localhost Attempt to verify that your session is now on the stndby database.

**Note:** During switchover, you lost the exiting session.

SQL> **select instance\_name from v$instance;**

select instance\_name from v$instance

\*

ERROR at line 1:

ORA-03113: end-of-file on communication channel Process ID: 24566

Session ID: 297 Serial number: 59237

1. Establish a new session using the PRMY.EXAMPLE.COM service.

SQL> **connect system/<password>@prmy**

Connected.

SQL>

1. Verify that you are now connected to the stndby instance (the primary database).

SQL> **select instance\_name from v$instance;**

INSTANCE\_NAME

stndby

1. Verify that your connection has been established with the DEV1 pluggable database and not the root container. Exit SQL\*Plus.

SQL> **select sys\_context ('USERENV', 'CON\_NAME') as container FROM dual;**

CONTAINER DEV1

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0

[oracle@host02 ~]

1. Return to the DGMGRL session running on localhost in Step 10. Validate both databases are ready for switchover, and then perform a switchover to return the configuration to the way it was at the start of this practice.

DGMGRL> **validate database stndby** Database Role: Primary database Ready for Switchover: ***Yes***

DGMGRL> **validate database orclcdb**

Database Role: Physical standby database Primary Database: stndby

Ready for Switchover: ***Yes***

Ready for Failover: Yes (Primary Running) Current Log File Groups Configuration:

Thread # Online Redo Log Groups Standby Redo Log Groups (stndby)

1 3 2

Future Log File Groups Configuration:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thread # | Online Redo | Log | Groups | Standby Redo Log Groups  (stndby) |
| 1 | 3 |  |  | 2 |

DGMGRL> **switchover to orclcdb**

Performing switchover NOW, please wait...

Operation requires a connection to database "orclcdb" Connecting ...

Connected to "orclcdb" Connected as SYSDG.

New primary database "orclcdb" is opening...

Operation requires start up of instance "stndby" on database "stndby"

Starting instance "stndby"... Connected to an idle instance. ORACLE instance started.

Connected to "stndby" Database mounted.

Database opened. Connected to "stndby" Connected to "orclcdb"

Switchover succeeded, new primary is "orclcdb" DGMGRL>

1. Display the status of the data guard configuration. Wait until all warning messages are cleared for practice 18-2.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby - Physical standby database

stndby2 - Logical standby database (disabled) ORA-16749: The member was disabled manually.

Members Not Receiving Redo: stndbyFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:

SUCCESS (status updated 48 seconds ago)

DGMGRL>

1. Keep the DGMGRL session on localhost for practice 18-2.

## Practice 18-2: Modifying the Primary Database Service for Application Continuity

### Overview

In this practice, you will modify the service PRMY created in the DEV1 pluggable database on the primary database to support Application Continuity.

### Tasks

1. Open a terminal window logged in as oracle to localhost with the environment variables set for orclcdb appropriately. Run the setup18.sh script to prepare for a simple testing.

[oracle@localhost]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orclcdb**

The Oracle base has been set to /u01/app/oracle [oracle@localhost ~]$ **/home/oracle/setup/lab\_18-2/setup18.sh**

drop table emp

\* ERROR at line 1:

ORA-00942: table or view does not exist

Table created. [oracle@localhost ~]$

1. Launch SQL\*Plus and set the session container to the DEV1 pluggable database.

[oracle@localhost ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Thu Jun 11 22:30:48 2020

Version 19.3.0.0.0

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

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **alter session set container=dev1;**

Session altered.

1. Modify the service PRMY.EXAMPLE.COM to support Application Continuity. Exit SQL\*Plus.

#### SQL> @/home/oracle/setup/modify\_svc.sql

SQL> DECLARE

1. params dbms\_service.svc\_parameter\_array;
2. BEGIN
3. params('FAILOVER\_TYPE'):='TRANSACTION';
4. params('REPLAY\_INITIATION\_TIMEOUT'):=1800;
5. params('RETENTION\_TIMEOUT'):=86400;
6. params('FAILOVER\_DELAY'):=10;
7. params('FAILOVER\_RETRIES'):=30;
8. params('FAILOVER\_RESTORE'):='LEVEL1';
9. params('commit\_outcome'):='true';
10. params('aq\_ha\_notifications'):='true';
11. dbms\_service.modify\_service('prmy.example.com',params);
12. END;

14 /

PL/SQL procedure successfully completed.

SQL> **exit**

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 - Production Version 19.3.0.0.0 [oracle@localhost ~]$

**Note:** You have to set two mandatory service attributes (FAILOVER\_TYPE and

COMMIT\_OUTCOME) to enable the Application Continuity feature.

1. Examine the script that you will soon use to execute the practice application. The script executes the application code called actest.jar.

[oracle@localhost ~]$ **cat /home/oracle/setup/lab\_18-2/runreplay**

java -

classpath ./actest.jar:$ORACLE\_HOME/ucp/lib/ucp.jar:$ORACLE\_HOME/ jdbc/lib/ojdbc8.jar actest.ACTest actest\_replay.properties

[oracle@localhost ~]$

1. Examine the properties file.
   * Use the Replay Data source: oracle.jdbc.replay.OracleDataSourceImpl
   * Use the Application Continuity enabled service: prmy
   * Use the Application Continuity supported connection pool: UCP

[oracle@localhost ~]$

**cat /home/oracle/setup/lab\_18-2/actest\_replay.properties**

username=hr password=hr autoCommit=false

# Use new replay datasource datasource=oracle.jdbc.replay.OracleDataSourceImpl

url=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=h ost01.example.com)(PORT=1521)(ADDRESS=(PROTOCOL=tcp)(HOST=stndby. example.com)(PORT=1521))(CONNECT\_DATA=(SERVICE\_NAME=prmy)))

# UCP setting:

ucp\_pool\_size=2 ucp\_validate\_connection\_on\_borrow=true ucp\_connection\_wait\_timeout=60

# Think Time taken to process the results from the database. Time in milliseconds.

# -1 means no sleep. thread\_think\_time=20

# Number of concurrent threads running in the application # UCP is tuned to have MAX and MIN limit set to this number\_of\_threads=6

verbose=true [oracle@localhost ~]$

1. Use the terminal window connected to host02. Execute the practice application. While the application runs, a periodic status message is displayed.

[oracle@host02 ~]$ **cd /home/oracle/setup/lab\_18-2/** [oracle@host02 lab\_18-2]$ **./runreplay** ######################################################

Connecting to jdbc:oracle:thin:@(DESCRIPTION\_LIST=(DESCRIPTION=(ADDRESS=(PROTOC OL=tcp)(HOST=localhost.example.com)(PORT=1521))(CONNECT\_DATA=(SERVIC E\_NAME=PRMY.EXAMPLE.COM)))(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HO ST=stndby.example.com)(PORT=1521))(CONNECT\_DATA=(SERVICE\_NAME=PRM Y.EXAMPLE.COM))))

# of Threads : 6

UCP pool size : 2

Thread think time : 20 ms ######################################################

2 active connections, avg response time from db 6 ms

1 active connections, avg response time from db 5 ms

1 active connections, avg response time from db 5 ms

...

1. While the application continues to execute, return to the DGMGRL session on localhost. Check the status of the data guard configuration and wait until all warning messages are cleared.

DGMGRL> **show configuration**

Configuration - DRSolution

Protection Mode: MaxPerformance Members:

orclcdb - Primary database orclcdbFS - Far sync instance

stndby - Physical standby database

stndby2 - Logical standby database (disabled) ORA-16749: The member was disabled manually.

Members Not Receiving Redo: stndbyFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:

SUCCESS (status updated 48 seconds ago)

DGMGRL>

1. Verify that both databases are ready for switchover, and then perform a switchover.

DGMGRL> **validate database orclcdb**

Database Role:

Primary database

Ready for Switchover: Yes

...

DGMGRL> **validate database stndby**

Database Role: Physical standby database Primary Database: orclcdb

Ready for Switchover: Yes

Ready for Failover: Yes (Primary Running)

...

DGMGRL> **switchover to stndby**

Performing switchover NOW, please wait...

Operation requires a connection to database "stndby" Connecting ...

Connected to "stndby" Connected as SYSDG.

New primary database "stndby" is opening...

Operation requires start up of instance "orclcdb" on database "orclcdb"

Starting instance "orclcdb"... Connected to an idle instance. ORACLE instance started.

Connected to "orclcdb" Database mounted.

Database opened. Connected to "orclcdb"

Switchover succeeded, new primary is "stndby"

DGMGRL>

1. Return to the terminal session on host02. Press Ctrl+C to abort the application.

^C[oracle@host02 lab\_18-2]$

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| …  2 | active | connections, | avg | response | time | from | db | 5 | ms |
| 2 | active | connections, | avg | response | time | from | db | 5 | ms |
| 0 | active | connections, | avg | response | time | from | db | 5 | ms |
| 2 | active | connections, | avg | response | time | from | db | 5 | ms |
| 1  2 | active active | connections, connections, | avg avg | response response | time time | from from | db db | 111 ms  5 ms | |
| 2 | active | connections, | avg | response | time | from | db | 7 | ms |
| 0 | active | connections, | avg | response | time | from | db | 6 | ms |
| 1 | active | connections, | avg | response | time | from | db | 5 | ms |
| 2 | active | connections, | avg | response | time | from | db | 5 | ms |
| … |  |  |  |  |  |  |  |  |  |

**Note:** Your application is still connected without having any connection issues. There was only a small delay during the switchover operation.

1. (**Optional**) Return to the DGMGRL session on localhost. Optionally, switch back to the original state.

DGMGRL> **switchover to orclcdb**

Performing switchover NOW, please wait...

Operation requires a connection to database "orclcdb" Connecting ...

Connected to "orclcdb" Connected as SYSDG.

New primary database "orclcdb" is opening...

Operation requires start up of instance "stndby" on database "stndby"

Starting instance "stndby"... Connected to an idle instance. ORACLE instance started.

Connected to "stndby" Database mounted.

Database opened. Connected to "stndby" Connected to "orclcdb"

Switchover succeeded, new primary is "orclcdb" DGMGRL>

1. Exit DGMGRL on localhost.

# Practices for Lesson 19: Patching and Upgrading Databases in a Data Guard Configuration

## Practices for Lesson 19

There are no practices for this lesson.