

# Oracle Database 19c

## Data Guard Administration



---

# Table of Contents

<b>Course Practice Environment: Security Credentials.....</b>	<b>7</b>
<b>Practices for Lesson 1: Introduction to Oracle Data Guard.....</b>	<b>9</b>
Practices for Lesson 1: Overview .....	10
Practice 1-1: Discovering the Practice Environment .....	11
<b>Practices for Lesson 2: Oracle Net Services in a Data Guard Environment.....</b>	<b>23</b>
Practices for Lesson 2: Overview .....	24
Practice 2-1: Configuring the tnsnames.ora File.....	25
Practice 2-2: Configuring the listener.ora File.....	33
<b>Practices for Lesson 3: Creating a Physical Standby Database by Using SQL and RMAN Commands .....</b>	<b>41</b>
Practices for Lesson 3: Overview .....	42
Practice 3-1: Prepare the Primary Database to Support Data Guard .....	43
Practice 3-2: Prepare Host and Create Physical Standby Database .....	46
Practice 3-3: Start Redo Transport and Verify Operation.....	57
<b>Practices for Lesson 4: Managing Physical Standby Files After Structural Changes on the Primary Database .....</b>	<b>61</b>
Practices for Lesson 4: Overview .....	62
Practice 4-1: Refreshing the Password File .....	63
Practice 4-2: Controlling PDB Replication.....	67
Practice 4-3: Automating Instantiation of a PDB .....	71
<b>Practices for Lesson 5: Using Oracle Active Data Guard: Supported Workloads in Read-Only Standby.....</b>	<b>79</b>
Practices for Lesson 5: Overview .....	80
Practice 5-1: Enable Active Data Guard Real-Time Query .....	81
Practice 5-2: Performing DDL/DML on Global Temporary Table .....	87
Practice 5-3: Managing Private Temporary Table for DDL/DML.....	90
Practice 5-4: Configuring Automatic Redirection of DML operations .....	95
<b>Practices for Lesson 6: Using Oracle Active Data Guard: Far Sync and Real-Time Cascading.....</b>	<b>101</b>
Practices for Lesson 6: Overview .....	102
Practice 6-1: Add Far Sync to the Data Guard Environment.....	103
Practice 6-2: Add 2nd Far Sync to the Data Guard Environment .....	111
<b>Practices for Lesson 7: Creating and Managing a Snapshot Standby Database.....</b>	<b>117</b>
Practices for Lesson 7: Overview .....	118
Practice 7-1: Convert Physical Standby to a Snapshot Standby.....	119
Practice 7-2: Convert Snapshot Standby Back to Physical Standby.....	125
<b>Practices for Lesson 8: Creating a Logical Standby Database.....</b>	<b>129</b>

Practices for Lesson 8: Overview .....	130
Practice 8-1: Identify Unsupported Objects for Logical Standbys .....	131
Practice 8-2: Create a Logical Standby (Temporarily a Physical) .....	133
Practice 8-3: Start Redo Transport and Verify Operation .....	146
Practice 8-4: Convert Physical Standby to Logical Standby .....	149
<b>Practices for Lesson 9: Oracle Data Guard Broker: Overview.....</b>	<b>153</b>
Practices for Lesson 9 .....	154
<b>Practices for Lesson 10: Creating a Data Guard Broker Configuration .....</b>	<b>155</b>
Practices for Lesson 10: Overview .....	156
Practice 10-1: Establishing Local and Remote Connections with DGMGRL .....	157
Practice 10-2: Create and Enable a Data Guard Broker Configuration .....	161
Practice 10-3: Verify and Examine the Data Guard Environment .....	171
<b>Practices for Lesson 11: Monitoring a Data Guard Broker Configuration.....</b>	<b>179</b>
Practices for Lesson 11: Overview .....	180
Practice 11-1: Monitoring the Physical Standby Database .....	181
Practice 11-2: Examining Data Guard Log and Trace Files.....	188
Practice 11-3: Using the VALIDATE commands.....	192
<b>Practices for Lesson 12: Configuring Data Protection Modes.....</b>	<b>199</b>
Practices for Lesson 12: Overview .....	200
Practice 12-1: Examining the Maximum Availability Protection Mode .....	201
Practice 12-2: Examining the Maximum Protection Mode .....	209
<b>Practices for Lesson 13: Optimizing and Tuning a Data Guard Configuration .....</b>	<b>219</b>
Practices for Lesson 13: Overview .....	220
Practice 13-1: Configuring Network Compression of Redo Data .....	221
Practice 13-2: Generating AWR Report for an Active Data Guard Instance.....	225
Practice 13-3: Using ADDM for an Active Data Guard Instance .....	246
Practice 13-4: Using SQL Tuning Advisor for an Active Data Guard Instance .....	254
<b>Practices for Lesson 14: Performing Role Transitions .....</b>	<b>265</b>
Practices for Lesson 14: Overview .....	266
Practice 14-1: Performing Switchover .....	267
Practice 14-2: Keeping Physical Standby Session Connected During Role Transition .....	275
<b>Practices for Lesson 15: Using Flashback Database in a Data Guard Configuration .....</b>	<b>285</b>
Practices for Lesson 15: Overview .....	286
Practice 15-1: Configuring Flashback Database on the Primary Database .....	287
Practice 15-2: Configuring Flashback Database on the Physical Standby Database.....	290
Practice 15-3: Configuring Flashback Database on the Logical Standby Database.....	293
Practice 15-4: Testing Automatic Flashback of Standby Database .....	296
Practice 15-5: Performing Flashback of the Logical Standby Database.....	306
<b>Practices for Lesson 16: Enabling Fast-Start Failover .....</b>	<b>313</b>
Practices for Lesson 16: Overview .....	314

Practice 16-1: Configuring Fast-Start Failover in Observer-Only Mode .....	315
Practice 16-2: Enabling Fast-Start Failover.....	324
Practice 16-3: Testing Fast-Start Failover.....	327
Practice 16-4: Switchover to Reinstated Database.....	332
<b>Practices for Lesson 17: Backup and Recovery Considerations in an Oracle Data Guard Configuration.....</b>	<b>335</b>
Practices for Lesson 17: Overview .....	336
Practice 17-1: Enable Change Tracking on the Physical Standby Database .....	337
Practice 17-2: Creating a Recovery Manager Catalog.....	339
Practice 17-3: Registering Your Database in the Recovery Catalog .....	342
Practice 17-4: Configuring RMAN Parameters .....	346
Practice 17-5: Recovering a Data File on Your Primary Database Over the Network.....	349
Practice 17-6: Rolling Forward a Standby Database with One Command .....	357
<b>Practices for Lesson 18: Enhanced Client Connectivity in a Data Guard Environment.....</b>	<b>367</b>
Practices for Lesson 18: Overview .....	368
Practice 18-1: Creating and Testing Primary Database Services .....	369
Practice 18-2: Modifying the Primary Database Service for Application Continuity .....	379
<b>Practices for Lesson 19: Patching and Upgrading Databases in a Data Guard Configuration .....</b>	<b>387</b>
Practices for Lesson 19 .....	388



## Course Practice Environment: Security Credentials

---

Operating System Credentials		
Practice Environment	Username	Password
Gateway VM	root	Read the note above
	oracle	
em13c	root	Oracle
	oracle	
host01 ~ host04	root	oracle
	oracle	

For product-specific credentials used in this course, see the following table:

Product-Specific Credentials		
Product/Application	DB Username	Password
rcatcdb	sysman	cloud_4U
rcatpdb	rconwer	oracle_4U
ORCL and PDB1	Any user	oracle_4U
All DBs and PDBs in host<nn>	Any user	oracle_4U



# **Practices for Lesson 1: Introduction to Oracle Data Guard**

## **Practices for Lesson 1: Overview**

---

### **Practices Overview**

In this practice, you will be introduced to the laboratory environment used to support all the practices during this course.

## Practice 1-1: Discovering the Practice Environment

---

### Overview

In this practice, you learn how to use the laboratory environment that supports all the practices in this course.

The laboratory environment for this course consists of six VMs:

- **gateway**: VM to connect to all other five VMs for practices
- **em13c**: VM for EMCC13c preinstalled and configured
- **host01**: VM for Primary CDB Database (**boston**)
- **host02**: VM for Far Sync Instance (**bostonFS**)
- **host03**: VM for Physical (**london**) and Logical Standby Databases (**london2**)
- **host04**: VM for Far Sync Instance (**londonFS**)

Each VM is a logically separate server running a distinct version of Oracle Linux, along with Oracle Database 19c (19.3) software and components from Oracle Enterprise Manager Cloud Control 13c (13.3).

To access virtual machines, you first establish a graphical session, which is connected to a gateway VM. Your instructor will provide specific details for each student's gateway. From there, you will create terminal sessions as required and connect to virtual machines by using SSH as described in the tasks for this practice.

Please refer to the supplied Course Practice Environment: Security Credentials document for the passwords that you will require to use the laboratory environment.

### Tasks

1. Open a terminal window. Log in as the `root` user and restart the `dnsmasq` service to ensure viability and availability of the service for the Enterprise Manager Cloud Control 13c access in our practice environment. Switch to the `oracle` user when done.

```
[oracle@gateway ~]$ su -
Password: <password>
Last login: Sat Jun 13 18:47:51 UTC 2020 on pts/0
[root@gateway ~]# systemctl restart dnsmasq.service
[root@gateway ~]# systemctl status dnsmasq.service
\u25cf dnsmasq.service - DNS caching server.
   Loaded: loaded (/usr/lib/systemd/system/dnsmasq.service;
             enabled; vendor preset: disabled)
     Active: active (running) since Sat 2020-06-13 18:56:26 UTC;
             10s ago
   Main PID: 12693 (dnsmasq)
      CGroup: /system.slice/dnsmasq.service
              \u2514\u250012693 /usr/sbin/dnsmasq -k
```

```
Jun 13 18:56:26 gateway.example.com systemd[1]: Stopped DNS
caching server..
Jun 13 18:56:26 gateway.example.com systemd[1]: Started DNS
caching server..
Jun 13 18:56:26 gateway.example.com dnsmasq[12693]: started,
version 2.76 cac...
Jun 13 18:56:26 gateway.example.com dnsmasq[12693]: compile time
options: IPv...
Jun 13 18:56:26 gateway.example.com dnsmasq[12693]: reading
/etc/resolv.conf
Jun 13 18:56:26 gateway.example.com dnsmasq[12693]: using
nameserver 10.237.1...
Jun 13 18:56:26 gateway.example.com dnsmasq[12693]: using
nameserver 152.68.1...
Jun 13 18:56:26 gateway.example.com dnsmasq[12693]: read
/etc/hosts - 9 addre...
Hint: Some lines were ellipsized, use -l to show in full.
[root@gateway ~]# exit
logout
[oracle@gateway ~]$
```

2. Review the `/etc/hosts` file. You should see the output similar to the example displayed below. It shows that your environment is configured with six VMs, which you will use in upcoming practices.

```
[oracle@gateway ~]$ more /etc/hosts
...
192.0.2.1      gateway.example.com    gateway
192.0.2.11     host01.example.com   host01
192.0.2.12     host02.example.com   host02
192.0.2.13     host03.example.com   host03
192.0.2.14     host04.example.com   host04
192.0.2.20     em13c.example.com    em13c  edvmr1p0.us.oracle.com
edvmr1p0

[oracle@gateway ~]$
```

3. Connect to host01 as the oracle OS user with the environment variables set for boston appropriately.

```
[oracle@gateway ~]$ ssh oracle@host01
oracle@host01's password: <password>

[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$
```

4. Using SQL\*Plus, connect to your database as the database administrator. Check the connection to the boston CDB database and verify that the status of PDB dev1 is open.

```
[oracle@host01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Mon Jun 1 10:06:26
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 name, cdb from v$database;

NAME      CDB
-----
BOSTON      YES

SQL> show pdbs

CON ID CON NAME          OPEN MODE  RESTRICTED
-----
2  PDB$SEED            READ ONLY  NO
3  DEV1                 READ WRITE NO
SQL>
```

5. Save the current PDB state to bring DEV1 to the saved state at the boston CDB startup.

```
SQL> alter pluggable database dev1 save state;
```

```
Pluggable database altered.
```

```
SQL>
```

6. Make sure that there is no value in the LOCAL\_LISTENER parameter.

```
SQL> show parameter local_listener
```

NAME	TYPE	VALUE
local_listener	string	

```
SQL>
```

7. Log in as the hr user and list the tables in the HR schema.

```
SQL> connect hr/<password>@host01:1521/dev1.example.com
```

```
Connected.
```

```
SQL> col tname format a30
```

```
SQL> select * from tab;
```

TNAME	TABTYPE	CLUSTERID
REGIONS	TABLE	
COUNTRIES	TABLE	
LOCATIONS	TABLE	
DEPARTMENTS	TABLE	
JOBS	TABLE	
EMPLOYEES	TABLE	
JOB_HISTORY	TABLE	
EMP_DETAILS_VIEW	VIEW	

```
8 rows selected.
```

```
SQL>
```

8. Log in as the oe user and list the tables in the OE schema. Exit your SQL\*Plus session.

```
SQL> connect oe/<password>@host01:1521/dev1.example.com
Connected.
SQL> col tname format a30
SQL> select * from tab;

TNAME          TABTYPE      CLUSTERID
-----
LINEORDER      TABLE
PART           TABLE
SUPPLIER       TABLE
CUSTOMER       TABLE
DATE_DIM       TABLE

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition
Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
[oracle@host01 ~]$
```

9. Connect to host02 as the oracle OS user. Set the environment variables by using the oraenv script. The /etc/oratab file has been updated for this course.

```
[oracle@host01 ~]$ ssh oracle@host02
oracle@host02's password: <password>
Last login: Mon Jun  8 01:11:16 2020 from host01.example.com
[oracle@host02 ~]$ . oraenv
ORACLE_SID = [oracle] ? bostonFS
The Oracle base has been set to /u01/app/oracle
```

10. Connect to host03 as the oracle OS user. Set the environment variables by using the oraenv script. The /etc/oratab file has been updated for this course.

```
[oracle@host02 ~]$ ssh oracle@host03
oracle@host03's password: <password>
Last login: Mon Jun  8 01:11:16 2020 from host01.example.com
[oracle@host02 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been set to /u01/app/oracle
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [london] ? london2
```

```
The Oracle base remains unchanged with value /u01/app/oracle  
[oracle@host03 ~]$
```

11. Connect to host04 as the oracle OS user. Set the environment variables by using the oraenv script. The /etc/oratab file has been updated for this course.

```
[oracle@host03 ~]$ ssh oracle@host04  
oracle@host03's password: <password>  
Last login: Mon Jun  8 01:11:16 2020 from host01.example.com  
[oracle@host02 ~]$ . oraenv  
ORACLE_SID = [oracle] ? londonFS  
The Oracle base has been set to /u01/app/oracle  
[oracle@host04 ~]$
```

12. Open a new terminal window connected to em13c as the oracle OS user.

```
[oracle@gateway ~]$ ssh oracle@em13c  
oracle@em13c's password: <Password>  
Last login: Mon Jun  8 00:03:16 2020 from gateway.example.com  
[oracle@em13c ~]$
```

13. Use the nslookup command to confirm that the DNS correctly resolves the EM host name (em13c.example.com) and its alias (edvmr1p0.us.oracle.com).

```
[oracle@em13c ~]$ nslookup em13c.example.com  
Server:          192.0.2.1  
Address:         192.0.2.1#53  
  
Name:   em13c.example.com  
Address: 192.0.2.20  
  
[oracle@em13c ~]$ nslookup edvmr1p0.us.oracle.com  
Server:          192.0.2.1  
Address:         192.0.2.1#53  
  
Name:   edvmr1p0.us.oracle.com  
Address: 192.0.2.20  
  
[oracle@em13c ~]$
```

14. Verify that the ORCL database and its PDB are running.

```
[oracle@em13c ~]$ sqlplus / as sysdba  
  
SQL*Plus: Release 19.0.0.0.0 - Production on Mon Jun  8 02:44:27  
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
----- -----
      2 PDB$SEED           READ ONLY NO
      3 PDB1                READ WRITE NO

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition
Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
[oracle@em13c ~]$
```

15. Verify that the RCATCDB database and its PDBs are running. Exit SQL\*Plus.

```

[oracle@em13c ~]$ . oraenv
ORACLE_SID = [ORCL] ? rcatcdb
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@em13c ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Mon Jun 8 02:45:39
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> show pdbs

  CON_ID CON_NAME          OPEN MODE  RESTRICTED
----- -----
      2 PDB$SEED           READ ONLY NO
      3 RCATPDB             READ WRITE NO
      4 EMCCPDB              READ WRITE NO

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition
Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
[oracle@em13c ~]$
```

16. Verify that the Oracle Management Server and Oracle Agent are running on em13c.

```
[oracle@em13c ~]$ /u01/app/em13c/omshome/bin/emctl status oms
Oracle Enterprise Manager Cloud Control 13c Release 3
(c) 1996, 2018 Oracle Corporation. All rights reserved.
WebTier is Up
Oracle Management Server is Up
JVMD Engine is Up
BI Publisher Server is Up
[oracle@em13c]$ /u01/app/em13c/agenthome/agent_13.3.0.0.0/bin/em
ctl status agent
Oracle Enterprise Manager Cloud Control 13c Release 3
(c) 1996, 2018 Oracle Corporation. All rights reserved.
-----
Agent Version          : 13.3.0.0.0
OMS Version           : 13.3.0.0.0
Protocol Version      : 12.1.0.1.0
Agent Home            : /u01/app/em13c/agenthome/agent_inst
Agent Log Directory   :
/u01/app/em13c/agenthome/agent_inst/sysman/log
Agent Binaries         :
/u01/app/em13c/agenthome/agent_13.3.0.0.0
Core JAR Location     :
/u01/app/em13c/agenthome/agent_13.3.0.0.0/jlib
Agent Process ID      : 2058
Parent Process ID     : 1901
Agent URL             :
https://edvmr1p0.us.oracle.com:3872/emd/main/
Local Agent URL in NAT :
https://edvmr1p0.us.oracle.com:3872/emd/main/
Repository URL        :
https://edvmr1p0.us.oracle.com:4903/empbs/upload
Started at             : 2020-06-03 03:30:07
Started by user        : oracle
Operating System       : Linux version 4.14.35-
1902.3.1.el7uek.x86_64 (amd64)
Number of Targets      : 43
Last Reload            : (none)
Last successful upload  : 2020-06-08
02:33:56
Last attempted upload   : 2020-06-08
02:33:56
Total Megabytes of XML files uploaded so far : 12.14
Number of XML files pending upload    : 0
```

```

Size of XML files pending upload(MB) : 0
Available disk space on upload filesystem : 21.31%
Collection Status : Collections
enabled

Heartbeat Status : Ok
Last attempted heartbeat to OMS : 2020-06-08
02:38:54
Last successful heartbeat to OMS : 2020-06-08
02:38:54
Next scheduled heartbeat to OMS : 2020-06-08
02:39:54

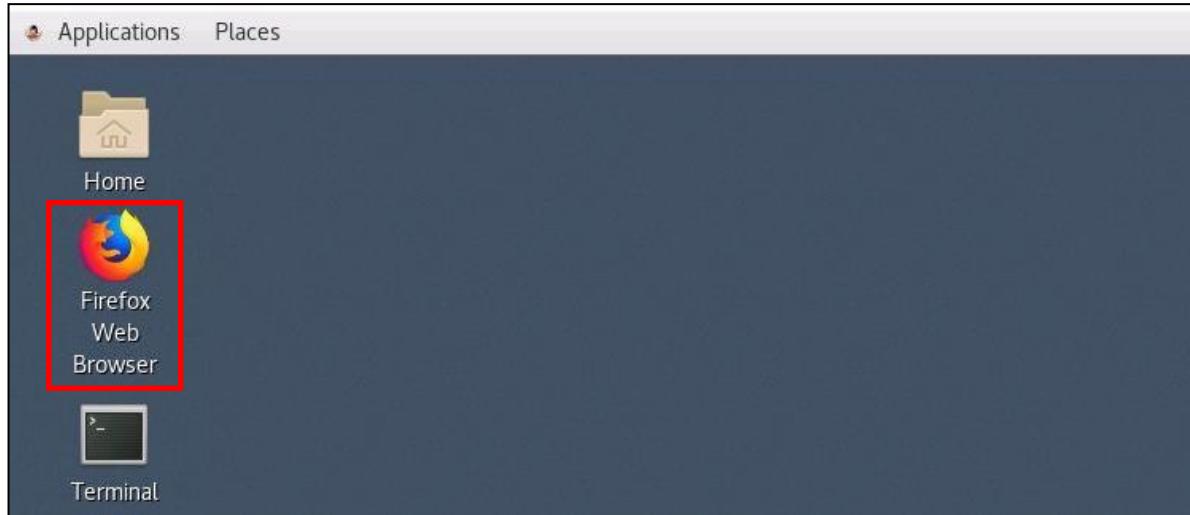
-----
Agent is Running and Ready
[oracle@em13c ~]$
```

17. **(Optional)** If the Oracle Management Server and Oracle Agent are not running, start with the following sequence of commands in the current terminal session:

```

[oracle@em13c ~]$ . oraenv
ORACLE_SID = [ORCL] ? rcatcdb
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@em13c ~]$
[oracle@em13c ~]$ lsnrctl start
[oracle@em13c ~]$ sqlplus / as sysdba
SQL> startup
SQL> exit
[oracle@em13c ~]$ /u01/app/em13c/omshome/bin/emctl start oms
[oracle@em13c ~]$
/u01/app/em13c/agenthome/agent_13.3.0.0.0/bin/emctl start agent
[oracle@em13c ~]$
```

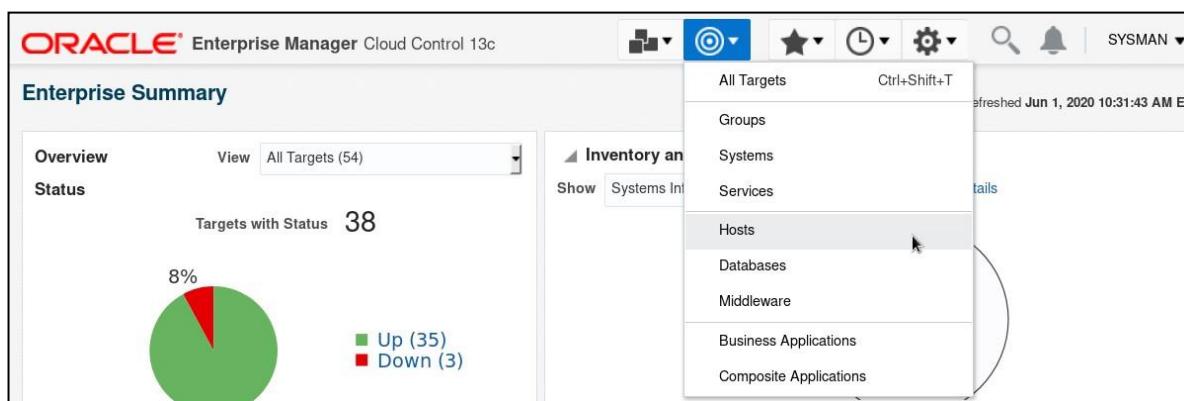
18. Using the Firefox Web Browser icon, start Firefox to access Enterprise Manager 13c.



19. Enter the URL for Cloud Control: In the current setup, use <https://em13c.example.com:7803/em>. Most probably, you receive a Secure Connection.
- At the end of the alert box, click **I Understand the Risks**.
  - At the bottom of the page, click **Add Exception**.
  - In the Add Security Exception pop-up window, click **Get Certificate**.
  - Confirm that “Permanently store this exception” is selected in your training environment and click **Confirm Security Exception**.

The Enterprise Manager Cloud Control console appears.

20. Enter **sysman** in the User Name field and the <password> in the Password field. Then click Login.
21. Click **Targets** and then **Hosts**, and verify that the agent is running on all five virtual machines.



Hosts											Page Refreshed Jun 8, 2020 2:52:56 AM UTC						
Search		Advanced Search															
View ▾		Add	Configure	Run Host Command													
Name	Status	Pending Activation	Incidents			Compliance Violations			Average Compliance Score	CPU Util %							
edvrmr1p0.us.oracle.com		-	0	0	0	0	0	0	-	30.5							
host01.example.com		-	0	0	0	0	0	0	-	0.33							
host02.example.com		-	0	0	0	0	0	0	-	0.14							
host03.example.com		-	0	0	0	0	0	0	-	0.14							
host04.example.com		-	0	0	0	0	0	0	-	0.1							

22. Click **Targets** and then **Databases**. On the Databases page, select Search List in the View option and expand the `boston.example.com` link and then “Pluggable Databases” to see the DEV1 PDB.

The screenshot shows two views of Oracle Enterprise Manager Cloud Control 13c:

- Enterprise Summary:** Shows an overview of 38 targets with 8% Down (3) and 92% Up (35). A pie chart indicates 8% Down and 92% Up. A dropdown menu is open under the Inventory navigation bar, showing options like All Targets, Groups, Systems, Services, Hosts, Databases (which is selected), Middleware, Business Applications, and Composite Applications.
- Databases:** Shows a list of databases including boston.example.com (Container), Pluggable Databases (DEV1), ORCL, and rcatcdb. The DEV1 database is expanded to show its Pluggable Database status. The table includes columns for Name, Type, Status, Target Version, Incidents, Average Compliance Score, and Member Status Summary.

Name	Type	Status	Target Version	Incidents	Average Compliance Score	Member Status Summary
boston.example.com	Database Instance : Container		19.3.0.0.0	0 0 0	N/A	0 1 0 0
Pluggable Databases		N/A		0 0 0	N/A	0 1 0 0
boston.example.com_DEV1	Pluggable Database		19.3.0.0.0	0 0 0	N/A	0 0 0 0
ORCL	Database Instance : Container		19.3.0.0.0	0 0 0	N/A	0 1 0 0
rcatcdb	Database Instance : Container		19.3.0.0.0	0 0 0	N/A	0 2 0 0

23. Click “Log Out” in the upper-right corner of the application and then exit enterprise Manager Cloud Control 13c when done.

The screenshot shows the Oracle Enterprise Manager Cloud Control 13c interface. The title bar reads "ORACLE Enterprise Manager Cloud Control 13c". The main menu bar includes "Databases", "Performance", "Availability", "Security", "Schema", "Administration", "Auto Refresh Off", "Help", "Welcome Page...", "Notification Center", "Entitlement Summary", "Set Current Page as My Home", "Personalize Page...", "Enterprise Manager Password & Email", "Preferences", "About Enterprise Manager", and "Log Out". The "Databases" section has tabs for "View", "Add", "Remove", and "Configure". A search bar at the top allows searching by Name. Below is a table with columns: Name, Type, Status, Target Version, and Incident. One row is visible for "boston.example.com" which is a Database Instance : Container. The status is green, target version is 19.3.0.0.0, and incident count is 0. The bottom status bar shows "Database Instance : Container" and "IN/A".

24. Close all terminal windows.

# **Practices for Lesson 2: Oracle Net Services in a Data Guard Environment**

## **Practices for Lesson 2: 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

In this practice, you will review the `tnsnames.ora` and `sqlnet.ora` configuration files, which include the net service names and settings used in this course. The following chart summarizes the entries that should be created:

Net Service Name	Host	Port	SDU	* _BUF_SIZE
BOSTON	host01.example.com	1521	65535	10485760
BOSTNFS	host02.example.com	1521	65535	10485760
LONDON	host03.example.com	1521	65535	10485760
LONDON2	host03.example.com	1521	65535	10485760
LONDONFS	host04.example.com	1521	65535	10485760
PRMY	host01.example.com host03.example.com	1521	65535	10485760
ORCL & PDB1	em13.example.com	1521	65535	10485760
RCATCDB & RCATPDB	em13.example.com	1521	65535	10485760

### Tasks

1. Open a terminal window and use the SSH client to connect to `host01` as the oracle OS user.

```
[oracle@gateway ~]$ ssh -X oracle@host01
oracle@host01's password: <password>
```

2. Use the `oraenv` utility to set the environment variables for the `boston` instance.

```
[oracle@host01]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
```

3. Create the `tnsnames.ora` configuration file by using the `host01-tnsnames.ora` file.

```
[oracle@host01 ~]$ cp /home/oracle/setup/host01-tnsnames.ora
$ORACLE_HOME/network/admin/tnsnames.ora
```

4. Create the `sqlnet.ora` configuration by using the `host01-sqlnet.ora` file.

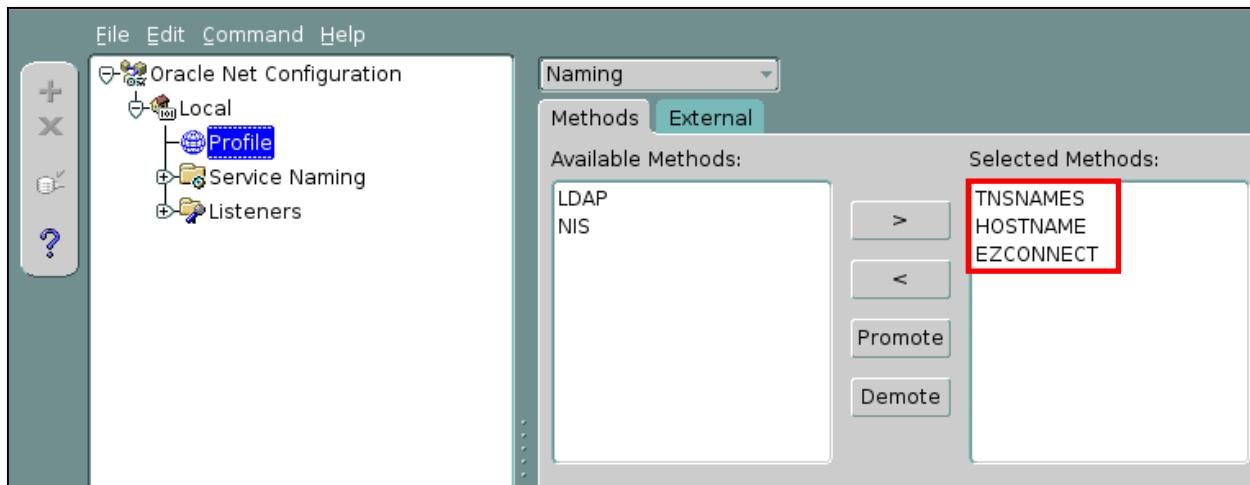
```
[oracle@host01 ~]$ cp /home/oracle/setup/host01-sqlnet.ora
$ORACLE_HOME/network/admin/sqlnet.ora
```

5. Invoke the Oracle Net Manager utility to review the `sqlnet.ora` file for Data Guard.

```
[oracle@host01 ~]$ netmgr
```

- Expand the **Local** menu tree item under Oracle Net Configuration
- Click the **Profile** menu tree item.

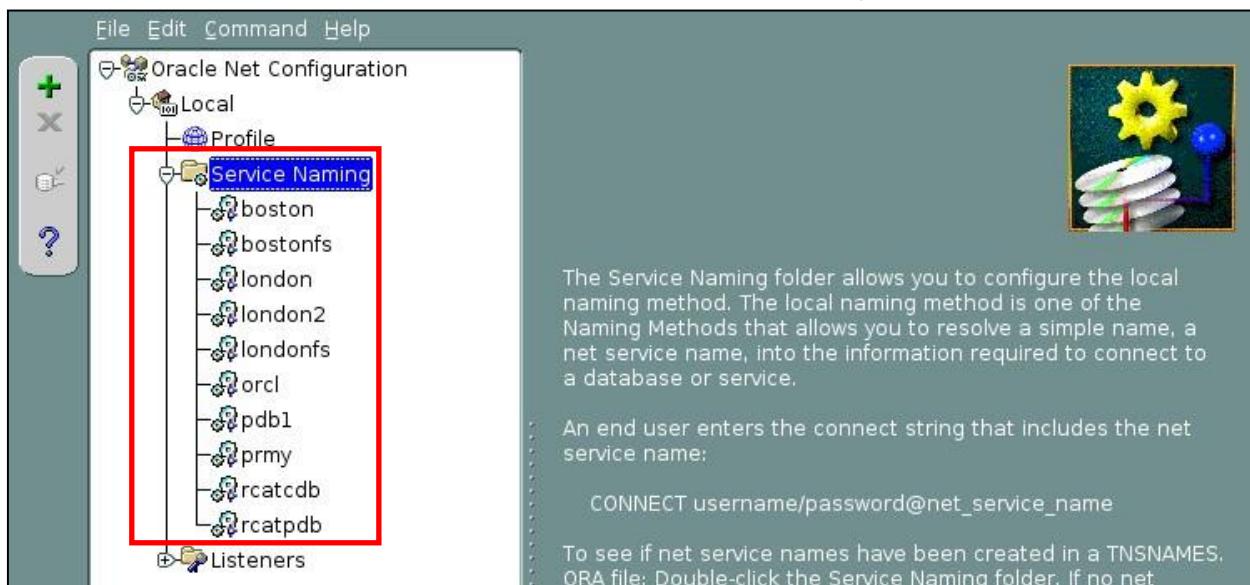
**Note:** You should see the EZCONNECT method under Selected Methods.



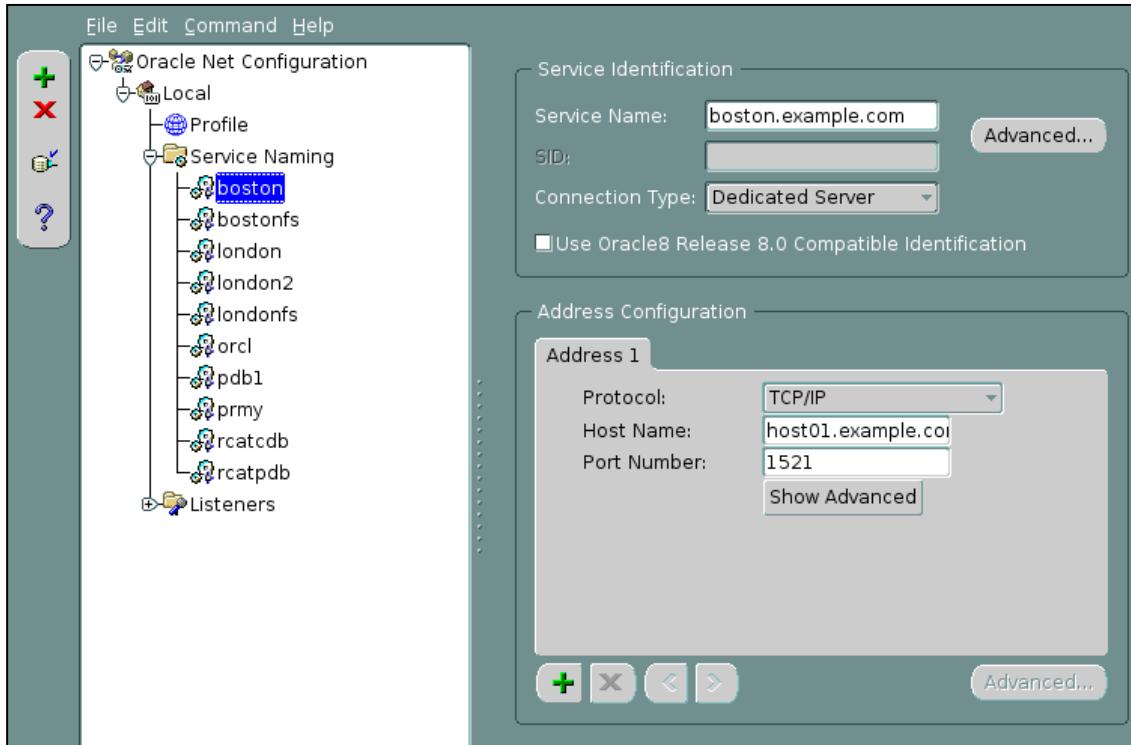
6. Using the Oracle Net Manager utility, review the `tnsnames.ora` file.

- Expand the **Service Naming** menu tree item.

**Note:** You will see the net service names used for future practices.

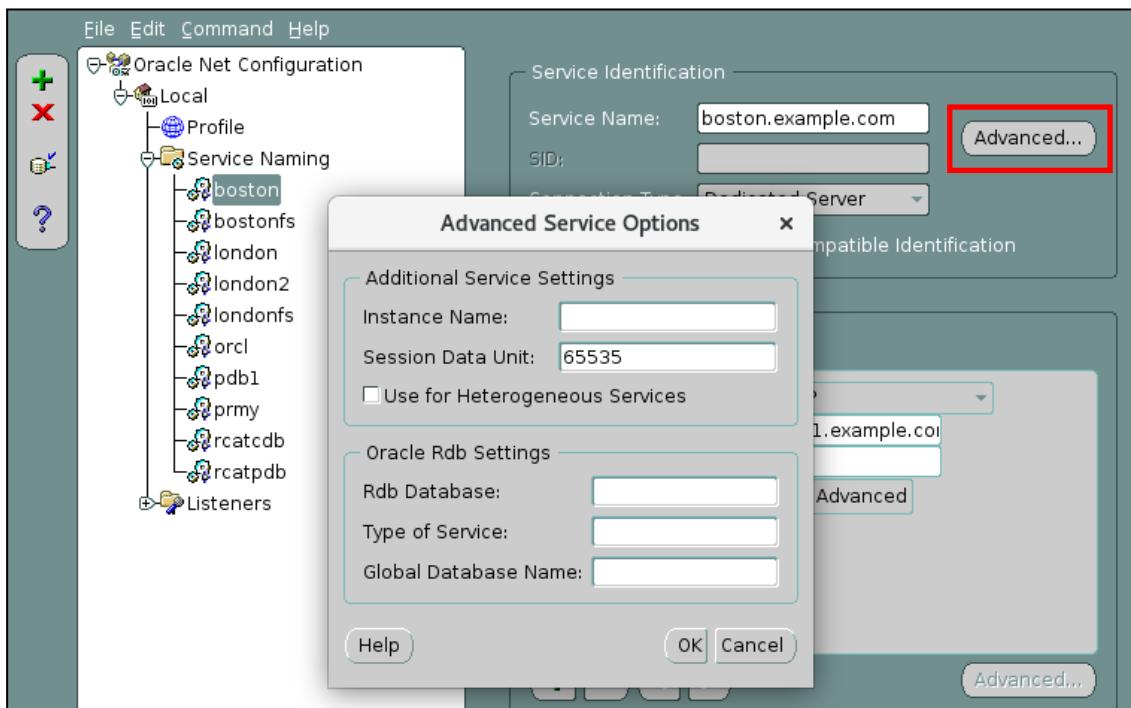


- b. Highlight each net service name (boston in this example) and review its configuration.



- c. Click the **Advanced** button in the Service Identification section.

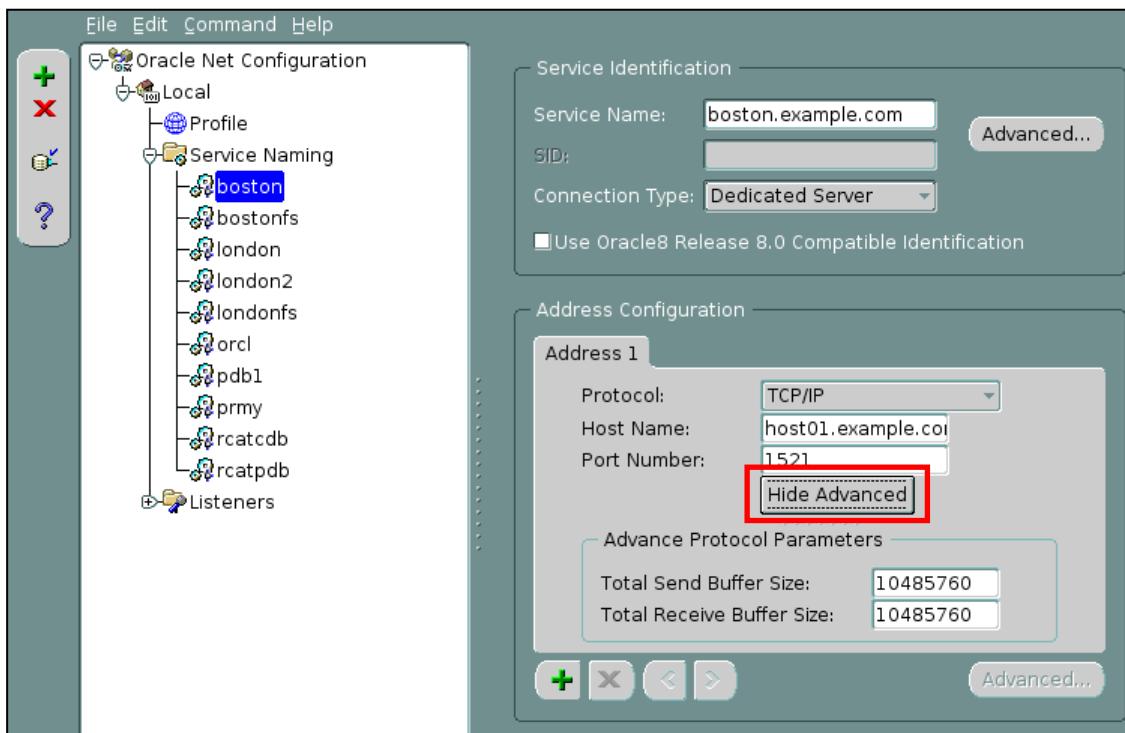
**Note:** The value of Session Data Unit should be **65535**.



- d. Click the **OK** button.

- e. Click the **Show Advanced** button in the Address Configuration section.

**Note:** The value of **Send Buffer Size** and **Receive Buffer Size** should be **10485760** in the Advanced Protocol Parameters section.



7. Click **File > Exit** from the menu.
8. Navigate to the `$ORACLE_HOME/network/admin` directory and examine the contents of the `sqlnet.ora` and `tnsnames.ora` network configuration files verifying the changes made.

```
[oracle@host01 ~]$ cd $ORACLE_HOME/network/admin
[oracle@host01 admin]$ cat sqlnet.ora
# sqlnet.ora Network Configuration File:
/u01/app/oracle/product/19.3.0/dbhome_1/network/admin/sqlnet.ora
# Generated by Oracle configuration tools.

NAMES.DIRECTORY_PATH= (TNSNAMES, HOSTNAME, EZCONNECT)

ADR_BASE = /u01/app/oracle


[oracle@host01 admin]$ cat tnsnames.ora
# tnsnames.ora Network Configuration File:
/u01/app/oracle/product/19.3.0/dbhome_1/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

BOSTON =
(DESCRIPTION =
```

```

(ADDRESS_LIST =
  (ADDRESS = (PROTOCOL = TCP) (HOST = host01.example.com) (PORT =
1521) (SEND_BUF_SIZE = 10485760) (RECV_BUF_SIZE = 10485760))
)
(SDU = 65535)
(CONNECT_DATA =
  (SERVER = DEDICATED)
  (SERVICE_NAME = boston.example.com)
)
)

BOSTONFS =
(DESCRIPTION =
(ADDRESS_LIST =
  (ADDRESS = (PROTOCOL = TCP) (HOST = host02.example.com) (PORT =
1521) (SEND_BUF_SIZE = 10485760) (RECV_BUF_SIZE = 10485760))
)
(SDU = 65535)
(CONNECT_DATA =
  (SERVICE_NAME = bostonFS.example.com)
)
)

LONDON =
(DESCRIPTION =
(ADDRESS_LIST =
  (ADDRESS = (PROTOCOL = TCP) (HOST = host03.example.com) (PORT =
1521) (SEND_BUF_SIZE = 10485760) (RECV_BUF_SIZE = 10485760))
)
(SDU = 65535)
(CONNECT_DATA =
  (SERVICE_NAME = london.example.com)
)
)

LONDON2 =
(DESCRIPTION =
(ADDRESS_LIST =
  (ADDRESS = (PROTOCOL = TCP) (HOST = host03.example.com) (PORT =
1521) (SEND_BUF_SIZE = 10485760) (RECV_BUF_SIZE = 10485760))
)
(SDU = 65535)
(CONNECT_DATA =
  (SERVICE_NAME = london2.example.com)
)
)
)

```

```

LONDONFS =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP) (HOST = host04.example.com) (PORT =
1521) (SEND_BUF_SIZE = 10485760) (RECV_BUF_SIZE = 10485760))
)
(SDU = 65535)
(CONNECT_DATA =
(SERVICE_NAME = londonFS.example.com)
)
)

PRMY =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP) (HOST = host01.example.com) (PORT =
1521) (SEND_BUF_SIZE = 10485760) (RECV_BUF_SIZE = 10485760))
(ADDRESS = (PROTOCOL = TCP) (HOST = host03.example.com) (PORT =
1521) (SEND_BUF_SIZE = 10485760) (RECV_BUF_SIZE = 10485760))
)
(SDU = 65535)
(CONNECT_DATA =
(SERVICE_NAME = prmy.example.com)
)
)

ORCL =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP) (HOST = em13c.example.com) (PORT =
1521) (SEND_BUF_SIZE = 10485760) (RECV_BUF_SIZE = 10485760))
)
(SDU = 65535)
(CONNECT_DATA =
(SERVICE_NAME = ORCL)
)
)

PDB1 =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP) (HOST = em13c.example.com) (PORT =
1521) (SEND_BUF_SIZE = 10485760) (RECV_BUF_SIZE = 10485760))
)
(SDU = 65535)
(CONNECT_DATA =
(SERVICE_NAME = pdb1)
)
)

```

```

        )
    )

RCATCDB =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP) (HOST = em13c.example.com) (PORT =
1521) (SEND_BUF_SIZE = 10485760) (RECV_BUF_SIZE = 10485760))
)
(SDU = 65535)
(CONNECT_DATA =
(SERVICE_NAME = rcatcdb)
)
)

RCATPDB =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP) (HOST = em13c.example.com) (PORT =
1521) (SEND_BUF_SIZE = 10485760) (RECV_BUF_SIZE = 10485760))
)
(SDU = 65535)
(CONNECT_DATA =
(SERVICE_NAME = rcatpdb)
)
)

LISTENER_ORCL =
(ADDRESS = (PROTOCOL = TCP) (HOST = em13c.example.com) (PORT = 1521))

LISTENER_BOSTON =
(ADDRESS = (PROTOCOL = TCP) (HOST = host01.example.com) (PORT = 1521))

LISTENER_BOSTONFS =
(ADDRESS = (PROTOCOL = TCP) (HOST = host02.example.com) (PORT = 1521))

LISTENER_LONDON =
(ADDRESS = (PROTOCOL = TCP) (HOST = host03.example.com) (PORT = 1521))

LISTENER_LONDON2 =
(ADDRESS = (PROTOCOL = TCP) (HOST = host03.example.com) (PORT = 1521))

LISTENER_LONDONFS =
(ADDRESS = (PROTOCOL = TCP) (HOST = host04.example.com) (PORT = 1521))

[oracle@host01 admin]$
```

9. Transfer the tnsnames.ora and sqlnet.ora configuration files to host02, host03, and host04.

```
[oracle@host01 admin]$ /home/oracle/setup/scp_tns.sh  
oracle@host02's password: <password>  
tnsnames.ora          100% 3495      3.9MB/s   00:00  
oracle@host03's password: <password>  
tnsnames.ora          100% 3495      3.7MB/s   00:00  
oracle@host04's password: <password>  
tnsnames.ora          100% 3495      3.5MB/s   00:00  
oracle@host02's password: <password>  
sqlnet.ora            100% 234       320.8KB/s  00:00  
oracle@host03's password: <password>  
sqlnet.ora            100% 234       252.4KB/s  00:00  
oracle@host04's password: <password>  
sqlnet.ora            100% 234       288.2KB/s  00:00  
[oracle@host01 admin]$
```

## Practice 2-2: Configuring the `listener.ora` File

---

### Overview

In this practice you will review the `listener.ora` configuration file in `host01`, which includes a static entry for RMAN to make a connection after shutting down the instance during physical or logical standby database creation. You will review the other entries that are required later in the course for Data Guard Broker operations. The following chart summarizes the entries that need to be created on each server host respectively:

Host	Global Database Name	Oracle Home Directory	SID
host01	boston.example.com	/u01/app/oracle/product/19.3.0/dbhome_1	boston
host01	boston_DGMGRL.example.com	/u01/app/oracle/product/19.3.0/dbhome_1	boston
host02	bostonFS.example.com	/u01/app/oracle/product/19.3.0/dbhome_1	bostonFS
host02	bostonFS_DGMGRL.example.com	/u01/app/oracle/product/19.3.0/dbhome_1	bostonFS
host03	london.example.com	/u01/app/oracle/product/19.3.0/dbhome_1	london
host03	london_DGMGRL.example.com	/u01/app/oracle/product/19.3.0/dbhome_1	london
host03	london2.example.com	/u01/app/oracle/product/19.3.0/dbhome_1	london2
host03	london2_DGMGRL.example.com	/u01/app/oracle/product/19.3.0/dbhome_1	london2
host04	londonFS.example.com	/u01/app/oracle/product/19.3.0/dbhome_1	londonFS
host04	londonFS_DGMGRL.example.com	/u01/app/oracle/product/19.3.0/dbhome_1	londonFS

### Tasks

1. If you closed the terminal window from the previous practice, open a terminal window and use the SSH client to connect to `host01` as the `oracle` OS user.

```
[oracle@gateway ~]$ ssh oracle@host01  
oracle@host01's password: <password>
```

2. Use the `oraenv` utility to set the environment variables for the `boston` instance.

```
[oracle@host01]$ . oraenv  
ORACLE_SID = [oracle] ? boston  
The Oracle base has been set to /u01/app/oracle
```

3. Replace the existing `listener.ora` file by using the `host01-listener.ora` file.

1. [oracle@host01 ~]\$ cp /home/oracle/setup/host01-listener.ora  
\$ORACLE\_HOME/network/admin/listener.ora
2. [oracle@host01 ~]\$

4. Navigate to the \$ORACLE\_HOME/network/admin directory and examine the contents of the listener.ora network configuration file.

```
[oracle@host01 ~]$ cd $ORACLE_HOME/network/admin
[oracle@host01 ~]$ cat listener.ora
# listener.ora Network Configuration File:
/u01/app/oracle/product/19.3.0/dbhome_1/network/admin/listener.ora
# Generated by Oracle configuration tools.

SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC =
(GLOBAL_DBNAME = boston.example.com)
(ORACLE_HOME = /u01/app/oracle/product/19.3.0/dbhome_1)
(SID_NAME = boston)
)
(SID_DESC =
(GLOBAL_DBNAME = boston_DGMGRL.example.com)
(ORACLE_HOME = /u01/app/oracle/product/19.3.0/dbhome_1)
(SID_NAME = boston)
)
)

LISTENER =
(DESCRIPTION_LIST =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP) (HOST =
host01.example.com) (PORT = 1521) (SEND_BUF_SIZE =
65535) (RECV_BUF_SIZE = 65535))
)
(DESCRIPTION =
(ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1521))
)
)

ADR_BASE_LISTENER = /u01/app/oracle
```

5. If the entries appear correct, reload the modified configuration for the listener that is now running. If the listener is not running, then start it instead of reloading it.

```
[oracle@host01 admin]$ lsnrctl reload or lsnrctl start
```

```
LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 07-JUN-2020 21:26:37
```

```
(c) 1991, 2019, Oracle. All rights reserved.
```

```
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=host01.example.com) (PORT=1521) (SEND_BUF_SIZE=10485760) (RECV_BUF_SIZE=10485760)))
The command completed successfully
[oracle@host01 admin]$
```

6. Create the listener.ora file in host02, host03, and host04 using the host<nn>-listener.ora file.

```
3. [oracle@host01 admin]$ /home/oracle/setup/scp_lsnr.sh
4. oracle@host02's password: <password>
5. host02-listener.ora          100%  709      18.6KB/s
   00:00
6. oracle@host03's password: <password>
7. host03-listener.ora          100% 1019      9.4KB/s
   00:00
8. oracle@host04's password: <password>
9. host04-listener.ora          100%  710     802.9KB/s
   00:00
10. [oracle@host01 admin]$
```

7. Open a new terminal window and use the SSH client to connect to host02 as the oracle OS user. You can leave open the previous window to host01. You will soon need windows open to host01, host02, host03, and host04.

```
[oracle@gateway ~]$ ssh oracle@host02
oracle@host02's password: <password>
```

8. Use the oraenv utility to set the environment variables.

```
[oracle@host02 ~]$ . oraenv
ORACLE_SID = [oracle] ? bostonFS
The Oracle base has been set to /u01/app/oracle
```

9. Navigate to the \$ORACLE\_HOME/network/admin directory and examine the contents of the listener.ora network configuration file verifying the changes made.

```
[oracle@host02 ~]$ cd $ORACLE_HOME/network/admin
[oracle@host02 ~]$ cat listener.ora
# listener.ora Network Configuration File:
/u01/app/oracle/product/19.3.0/dbhome_1/network/admin/listener.ora
# Generated by Oracle configuration tools.

SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC =
(GLOBAL_DBNAME = bostonFS.example.com)
(ORACLE_HOME = /u01/app/oracle/product/19.3.0/dbhome_1)
(SID_NAME = boston)
)
(SID_DESC =
(GLOBAL_DBNAME = bostonFS_DGMGRL.example.com)
(ORACLE_HOME = /u01/app/oracle/product/19.3.0/dbhome_1)
(SID_NAME = boston)
)
)

LISTENER =
(DESCRIPTION_LIST =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP) (HOST =
host02.example.com) (PORT = 1521) (SEND_BUF_SIZE =
65535) (RECV_BUF_SIZE = 65535))
)
(DESCRIPTION =
(ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1521))
)
)

ADR_BASE_LISTENER = /u01/app/oracle
```

10. If the entries appear correct, reload the modified configuration for the listener that is now running. If the listener is not running, then start it instead of reloading it.

```
[oracle@host02 admin]$ lsnrctl reload or lsnrctl start

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 09-JUN-
2020 13:41:14

(c) 1991, 2019, Oracle. All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=host02.example.com) (PO
RT=1521) (SEND_BUF_SIZE=10485760) (RECV_BUF_SIZE=10485760)))
The command completed successfully
[oracle@host02 admin]$
```

11. Open a new terminal window and use the SSH client to connect to host03 as the oracle OS user. You can leave open the previous windows to host01 and host02 open.

```
[oracle@gateway ~]$ ssh oracle@host03
oracle@host03's password: <password>
```

12. Use the oraenv utility to set the environment variables.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been set to /u01/app/oracle
```

13. Navigate to the \$ORACLE\_HOME/network/admin directory and examine the contents of the listener.ora network configuration file verifying the changes made.

```
[oracle@host03 ~]$ cd $ORACLE_HOME/network/admin
[oracle@host03 ~]$ cat listener.ora
# listener.ora Network Configuration File:
/u01/app/oracle/product/19.3.0/dbhome_1/network/admin/listener.o
ra
# Generated by Oracle configuration tools.

SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC =
(GLOBAL_DBNAME = london.example.com)
(ORACLE_HOME = /u01/app/oracle/product/19.3.0/dbhome_1)
(SID_NAME = boston)
)
(SID_DESC =
```

```

        (GLOBAL_DBNAME = london_DGMGRL.example.com)
        (ORACLE_HOME = /u01/app/oracle/product/19.3.0/dbhome_1)
        (SID_NAME = boston)
    )
    (SID_DESC =
        (GLOBAL_DBNAME = london2.example.com)
        (ORACLE_HOME = /u01/app/oracle/product/19.3.0/dbhome_1)
        (SID_NAME = boston)
    )
    (SID_DESC =
        (GLOBAL_DBNAME = london2_DGMGRL.example.com)
        (ORACLE_HOME = /u01/app/oracle/product/19.3.0/dbhome_1)
        (SID_NAME = boston)
    )
)

LISTENER =
(DESCRIPTION_LIST =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP) (HOST =
host03.example.com) (PORT = 1521) (SEND_BUF_SIZE =
65535) (RECV_BUF_SIZE = 65535))
)
(DESCRIPTION =
(ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1521))
)
)
ADR_BASE_LISTENER = /u01/app/oracle

```

14. If the entries appear correct, reload the modified configuration for the listener that is now running. If the listener is not running, then start it instead of reloading it.

```

[oracle@host03 admin]$ lsnrctl reload or lsnrctl start

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 09-JUN-
2020 13:42:05

(c) 1991, 2019, Oracle. All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=host03.example.com) (PO
RT=1521) (SEND_BUF_SIZE=10485760) (RECV_BUF_SIZE=10485760)))

```

```
The command completed successfully  
[oracle@host03 admin]$
```

15. Open a new terminal window and use the SSH client to connect to host04 as the oracle OS user. You can leave open the previous windows to host01, host02, and host03 open.

```
[oracle@gateway ~]$ ssh oracle@host04  
oracle@host04's password: <password>
```

16. Use the oraenv utility to set the environment variables.

```
[oracle@host04 ~]$ . oraenv  
ORACLE_SID = [oracle] ? londonFS  
The Oracle base has been set to /u01/app/oracle
```

17. Navigate to the \$ORACLE\_HOME/network/admin directory and examine the contents of the listener.ora network configuration file verifying the changes made.

```
[oracle@host04 ~]$ cd $ORACLE_HOME/network/admin  
[oracle@host04 admin]$ cat listener.ora  
# listener.ora Network Configuration File:  
/u01/app/oracle/product/19.3.0/dbhome_1/network/admin/listener.ora  
# Generated by Oracle configuration tools.  
  
SID_LIST_LISTENER =  
(SID_LIST =  
  (SID_DESC =  
    (GLOBAL_DBNAME = londonFS.example.com)  
    (ORACLE_HOME = /u01/app/oracle/product/19.3.0/dbhome_1)  
    (SID_NAME = boston)  
  )  
  (SID_DESC =  
    (GLOBAL_DBNAME = londonFS_DGMGRl.example.com)  
    (ORACLE_HOME = /u01/app/oracle/product/19.3.0/dbhome_1)  
    (SID_NAME = boston)  
  )  
)  
  
LISTENER =  
(DESCRIPTION_LIST =  
  (DESCRIPTION =
```

```

        (ADDRESS = (PROTOCOL = TCP) (HOST =
host04.example.com) (PORT = 1521) (SEND_BUF_SIZE =
65535) (RECV_BUF_SIZE = 65535))
    )
    (DESCRIPTION =
        (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1521))
    )
)

ADR_BASE_LISTENER = /u01/app/oracle

```

18. If the entries appear correct, reload the modified configuration for the listener that is now running. If the listener is not running, then start it instead of reloading it.

```
[oracle@host04 admin]$ lsnrctl reload or lsnrctl start

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 09-JUN-
2020 13:43:01

(c) 1991, 2019, Oracle. All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=host04.example.com) (PO-
RT=1521) (SEND_BUF_SIZE=10485760) (RECV_BUF_SIZE=10485760)))
The command completed successfully
[oracle@host04 admin]$
```

19. It is advised to keep one terminal window open to each of host01, host02, host03, and host04. If you should close any terminal windows, log in to each host as the oracle account and set the environment variables by using the oraenv utility as in previous tasks. For host03, you could open two distinct windows: one for london and the other for london2. The instance name to use with the oraenv utility depends on which host you are connected to according to the following chart:

<b>Host</b>	<b>oraenv Utility Instance Name</b>
host01	boston
host02	bostonFS
host03	london or london2
host04	londonFS

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

## **Practices for Lesson 3: Overview**

---

### **Practices Overview**

In these practices, you will prepare `host01` to become the primary database and `host03` to become a physical standby database. You will use the `RMAN` utility to create the physical standby database and then verify its operation.

## 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.

### Tasks

1. On host01, invoke SQL\*Plus and connect as SYSDBA to your primary database (boston). Determine if FORCE LOGGING is enabled. If it is not enabled, enable FORCE LOGGING mode.

```
[oracle@host01 ~]$ 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
```

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> @/home/oracle/setup/add_srl.sql
SQL> alter database add standby logfile
  ('/u01/app/oracle/oradata/BOSTON/stdbyredo01.log') size 200M;

Database altered.

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

Database altered.

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

Database altered.

SQL> alter database add standby logfile
  ('/u01/app/oracle/oradata/BOSTON/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;
```

```
4    209715200  
5    209715200  
6    209715200  
7    209715200
```

3. 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> @/home/oracle/setup/set_LAD_1.sql  
SQL> alter system set  
log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST  
valid_for=(ALL_LOGFILES,ALL_ROLES) db_unique_name=boston'  
scope=both;  
  
System altered.  
  
SQL> alter system set log_archive_dest_state_1='enable'  
scope=both;  
  
System altered.
```

4. 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.

5. Define the `log_archive_config` parameter to include entries for: boston, bostonFS, london, londonFS, and london2. Only london is needed at this moment, but the others can be added now in preparation for upcoming practices.

```
SQL> alter system set  
log_archive_config='dg_config=(boston,bostonFS,london,londonFS,  
london2)' scope=both;  
System altered.
```

6. 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 host03 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 `host03` to create the initial directories needed for a physical standby database by running the `crdir_host03` script. Since the primary database is using multi-tenant architecture, additional directories are needed.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been set to /u01/app/oracle
[oracle@host03 ~]$ cat /home/oracle/setup/crdir_host03.sh
...
mkdir -p /u01/app/oracle/admin/london/adump
mkdir -p /u01/app/oracle/oradata/LONDON
mkdir -p /u01/app/oracle/oradata/LONDON/pdbseed
mkdir -p /u01/app/oracle/oradata/LONDON/dev1
mkdir -p /u01/app/oracle/fast_recovery_area/LONDON
[oracle@host03 ~]$ /home/oracle/setup/crdir_host03.sh
[oracle@host03 ~]$
```

2. Create a starter initialization file on `host03` for the `london` physical standby instance by using the `create_init.sh` script. The only two parameters required are `DB_NAME` and `DB_DOMAIN`.

```
[oracle@host03 ~]$ cat /home/oracle/setup/create_init.sh
...
echo 'DB_NAME=london' >
/u01/app/oracle/product/19.3.0/dbhome_1/dbs/initlondon.ora
echo 'DB_DOMAIN=example.com' >>
/u01/app/oracle/product/19.3.0/dbhome_1/dbs/initlondon.ora
[oracle@host03 ~]$ /home/oracle/setup/create_init.sh
[oracle@host03 ~]$
```

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@host03 ~]$ /home/oracle/setup/scp_pwdfile.sh
The authenticity of host 'host01 (192.0.2.11)' 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 'host01,192.0.2.11' (ECDSA) to the
list of known hosts.
oracle@host01's password: <password>
orapwboston                               100% 2048      2.8MB/s   00:00
[oracle@host03 ~]$
```

- Start up nomount the london standby instance on host03. This assumes that the terminal window you are using has previously set the environment variables to london. Exit SQL\*Plus when done.

```

[oracle@host03 ~]$ 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
```

- Review the create\_london.rman script.

```
[oracle@host03 ~]$ cat /home/oracle/setup/create_london.rman
run {
allocate channel prmy1 type disk;
allocate auxiliary channel stby1 type disk;
```

```

duplicate target database for standby from active database
spfile
parameter_value_convert ('boston','london','BOSTON','LONDON')
  set db_name='boston'
  set db_unique_name='london'
  set db_file_name_convert='BOSTON','LONDON'
  set log_file_name_convert='BOSTON','LONDON'
  set fal_server='boston'
  set log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST
    valid_for=(ALL_LOGFILES,ALL_ROLES) db_unique_name=london'
nofilenamecheck;
allocate auxiliary channel stby type disk;
sql channel stby "alter database recover managed standby
database disconnect"; }

```

6. Create a physical standby on host03 by using the RMAN utility.

```

[oracle@host03 ~]$ rman target sys/<password>@boston auxiliary
sys/<password>@london
cmdfile=/home/oracle/setup/create_london.rman

Recovery Manager: Release 19.0.0.0.0 - Production on Tue Jun 2
16:30:27 2020
Version 19.3.0.0.0

(c) 1982, 2019, Oracle and/or its affiliates. Allrights
reserved.

connected to target database: BOSTON (DBID=2732274290)
connected to auxiliary database: LONDON (not mounted)

RMAN> run {
2> allocate channel prmy1 type disk;
3> allocate auxiliary channel stby1 type disk;
4> duplicate target database for standby from active database
5> spfile
6> parameter_value_convert ('boston','london')
7>   set db_name='boston'
8>   set db_unique_name='london'
9>   set db_file_name_convert='BOSTON','LONDON'
10>    set log_file_name_convert='BOSTON','LONDON'
11>    set fal_server='boston'

```

```

12>   set log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST
valid_for=(ALL_LOGFILES,ALL_ROLES) db_unique_name=london'
13> nofilenamecheck;
14> allocate auxiliary channel stby type disk;
15> sql channel stby "alter database recover managed standby
database disconnect"; }
16>
using target database control file instead of recovery catalog
allocated channel: prmy1
channel prmy1: SID=292 device type=DISK

allocated channel: stby1
channel stby1: SID=21 device type=DISK

Starting Duplicate Db at 02-JUN-20

contents of Memory Script:
{
  backup as copy reuse
  passwordfile auxiliary format
'/u01/app/oracle/product/19.3.0/dbhome_1/dbs/orapwlondon' ;
  restore clone from service 'boston' spfile to
'/u01/app/oracle/product/19.3.0/dbhome_1/dbs/spfilelondon.ora';
  sql clone "alter system set spfile=
''/u01/app/oracle/product/19.3.0/dbhome_1/dbs/spfilelondon.ora''";
}
executing Memory Script

Starting backup at 02-JUN-20
Finished backup at 02-JUN-20

Starting restore at 02-JUN-20

channel stby1: starting datafile backup set restore
channel stby1: using network backup set from service boston
channel stby1: restoring SPFILE
output file
name=/u01/app/oracle/product/19.3.0/dbhome_1/dbs/spfilelondon.or
a
channel stby1: restore complete, elapsed time: 00:00:01
Finished restore at 02-JUN-20

```

```

sql statement: alter system set spfile=
  ''/u01/app/oracle/product/19.3.0/dbhome_1/dbs/spfilelondon.ora'''

contents of Memory Script:
{
    sql clone "alter system set audit_file_dest =
  ''/u01/app/oracle/admin/london/adump'' comment=
  ''' scope=spfile";
    sql clone "alter system set dispatchers =
  '(PROTOCOL=TCP) (SERVICE=londonXDB)' comment=
  ''' scope=spfile";
    sql clone "alter system set db_name =
  'boston' comment=
  ''' scope=spfile";
    sql clone "alter system set db_unique_name =
  'london' comment=
  ''' scope=spfile";
    sql clone "alter system set db_file_name_convert =
  'BOSTON', 'LONDON' comment=
  ''' scope=spfile";
    sql clone "alter system set log_file_name_convert =
  'BOSTON', 'LONDON' comment=
  ''' scope=spfile";
    sql clone "alter system set fal_server =
  'boston' 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=london' comment=
  ''' scope=spfile";
    shutdown clone immediate;
    startup clone nomount;
}
executing Memory Script

sql statement: alter system set audit_file_dest =
  ''/u01/app/oracle/admin/london/adump'' comment= '''
  scope=spfile

sql statement: alter system set dispatchers = '(PROTOCOL=TCP)
  (SERVICE=londonXDB)' comment= ''' scope=spfile

```

```

sql statement: alter system set db_name = ''boston'' comment=
'''' scope=spfile

sql statement: alter system set db_unique_name = ''london''
comment= '''' scope=spfile

sql statement: alter system set db_file_name_convert =
'BOSTON', 'LONDON' comment= '''' scope=spfile

sql statement: alter system set log_file_name_convert =
'BOSTON', 'LONDON' comment= '''' scope=spfile

sql statement: alter system set fal_server = ''boston''
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=london''
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=256 device type=DISK

contents of Memory Script:
{
    restore clone from service 'boston' standby controlfile;
}
executing Memory Script

Starting restore at 02-JUN-20

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

```

```

channel stby1: restoring control file
channel stby1: restore complete, elapsed time: 00:00:02
output file name=/u01/app/oracle/oradata/BOSTON/control01.ctl
output file
name=/u01/app/oracle/fast_recovery_area/BOSTON/control02.ctl
Finished restore at 02-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/LONDON/temp01.dbf";
    set newname for tempfile 2 to
"/u01/app/oracle/oradata/LONDON/pdbseed/temp012020-06-02_14-25-
16-052-PM.dbf";
    set newname for tempfile 3 to
"/u01/app/oracle/oradata/LONDON/dev1/temp01.dbf";
    switch clone tempfile all;
    set newname for datafile 1 to
"/u01/app/oracle/oradata/LONDON/system01.dbf";
    set newname for datafile 3 to
"/u01/app/oracle/oradata/LONDON/sysaux01.dbf";
    set newname for datafile 4 to
"/u01/app/oracle/oradata/LONDON/undotbs01.dbf";
    set newname for datafile 5 to
"/u01/app/oracle/oradata/LONDON/pdbseed/system01.dbf";
    set newname for datafile 6 to
"/u01/app/oracle/oradata/LONDON/pdbseed/sysaux01.dbf";
    set newname for datafile 7 to
"/u01/app/oracle/oradata/LONDON/users01.dbf";
    set newname for datafile 8 to
"/u01/app/oracle/oradata/LONDON/pdbseed/undotbs01.dbf";
    set newname for datafile 9 to
"/u01/app/oracle/oradata/LONDON/dev1/system01.dbf";
    set newname for datafile 10 to
"/u01/app/oracle/oradata/LONDON/dev1/sysaux01.dbf";

```

```
      set newname for datafile 11 to
"/u01/app/oracle/oradata/LONDON/dev1/undotbs01.dbf";
      set newname for datafile 12 to
"/u01/app/oracle/oradata/LONDON/dev1/users01.dbf";
      restore
      from nonsparse    from service
'boston'   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/LONDON/temp01.dbf
in control file
renamed tempfile 2 to
/u01/app/oracle/oradata/LONDON/pdbseed/temp012020-06-02_14-25-
16-052-PM.dbf in control file
renamed tempfile 3 to
/u01/app/oracle/oradata/LONDON/dev1/temp01.dbf in control file

executing command: SET NEWNAME

executing command: SET NEWNAME
```

```
executing command: SET NEWNAME

executing command: SET NEWNAME

Starting restore at 02-JUN-20

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 stby1: starting datafile backup set restore
channel stby1: using network backup set from service boston
channel stby1: specifying datafile(s) to restore from backup set
channel stby1: restoring datafile 00001 to
/u01/app/oracle/oradata/LONDON/system01.dbf
channel stby1: restore complete, elapsed time: 00:00:35
channel stby1: starting datafile backup set restore
channel stby1: using network backup set from service boston
channel stby1: specifying datafile(s) to restore from backup set
channel stby1: restoring datafile 00003 to
/u01/app/oracle/oradata/LONDON/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 boston
channel stby1: specifying datafile(s) to restore from backup set
channel stby1: restoring datafile 00004 to
/u01/app/oracle/oradata/LONDON/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 boston
channel stby1: specifying datafile(s) to restore from backup set
channel stby1: restoring datafile 00007 to
/u01/app/oracle/oradata/LONDON/users01.dbf
channel stby1: restore complete, elapsed time: 00:00:01
channel stby1: starting datafile backup set restore
channel stby1: using network backup set from service boston
channel stby1: specifying datafile(s) to restore from backup set
channel stby1: restoring datafile 00009 to
/u01/app/oracle/oradata/LONDON/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 boston
channel stby1: specifying datafile(s) to restore from backup set
```

```

channel stby1: restoring datafile 00010 to
/u01/app/oracle/oradata/LONDON/dev1/sysaux01.dbf
channel stby1: restore complete, elapsed time: 00:00:08
channel stby1: starting datafile backup set restore
channel stby1: using network backup set from service boston
channel stby1: specifying datafile(s) to restore from backup set
channel stby1: restoring datafile 00011 to
/u01/app/oracle/oradata/LONDON/dev1/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 boston
channel stby1: specifying datafile(s) to restore from backup set
channel stby1: restoring datafile 00012 to
/u01/app/oracle/oradata/LONDON/dev1/users01.dbf
channel stby1: restore complete, elapsed time: 00:00:01
Finished restore at 02-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=4 STAMP=1042043611 file
name=/u01/app/oracle/oradata/LONDON/system01.dbf
datafile 3 switched to datafile copy
input datafile copy RECID=5 STAMP=1042043611 file
name=/u01/app/oracle/oradata/LONDON/sysaux01.dbf
datafile 4 switched to datafile copy
input datafile copy RECID=6 STAMP=1042043611 file
name=/u01/app/oracle/oradata/LONDON/undotbs01.dbf
datafile 5 switched to datafile copy
input datafile copy RECID=7 STAMP=1042043611 file
name=/u01/app/oracle/oradata/LONDON/pdbseed/system01.dbf
datafile 6 switched to datafile copy
input datafile copy RECID=8 STAMP=1042043611 file
name=/u01/app/oracle/oradata/LONDON/pdbseed/sysaux01.dbf
datafile 7 switched to datafile copy
input datafile copy RECID=9 STAMP=1042043611 file
name=/u01/app/oracle/oradata/LONDON/users01.dbf
datafile 8 switched to datafile copy

```

```
input datafile copy RECID=10 STAMP=1042043611 file
name=/u01/app/oracle/oradata/LONDON/pdbseed/undotbs01.dbf
datafile 9 switched to datafile copy
input datafile copy RECID=11 STAMP=1042043611 file
name=/u01/app/oracle/oradata/LONDON/dev1/system01.dbf
datafile 10 switched to datafile copy
input datafile copy RECID=12 STAMP=1042043611 file
name=/u01/app/oracle/oradata/LONDON/dev1/sysaux01.dbf
datafile 11 switched to datafile copy
input datafile copy RECID=13 STAMP=1042043611 file
name=/u01/app/oracle/oradata/LONDON/dev1/undotbs01.dbf
datafile 12 switched to datafile copy
input datafile copy RECID=14 STAMP=1042043611 file
name=/u01/app/oracle/oradata/LONDON/dev1/users01.dbf
Finished Duplicate Db at 02-JUN-20

allocated channel: stby
channel stby: SID=25 device type=DISK

sql statement: alter database recover managed standby database
disconnect
released channel: prmy1
released channel: stby1
released channel: stby

Recovery Manager complete.
```

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

---

### Overview

In this practice, you will start the redo transport from host01 to host03 and verify operation.

### Tasks

1. Use a terminal window for host01 logged in as oracle with the environment variables set to boston and start redo transport by defining log\_archive\_dest\_2 pointing to the physical standby database.

```
[oracle@host01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Mon Jun 1 16:56:54
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> @/home/oracle/setup/set_LAD_2.sql
SQL> alter system set log_archive_dest_2='SERVICE=london ASYNC
REOPEN=15 valid_for=(ONLINE_LOGFILES,PRIMARY_ROLE)
db_unique_name=london' scope=both;

System altered.

SQL>
```

2. Determine the last sequence number archived on the primary database (host01). Your number may be different in lab.

```
SQL> SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED_LOG GROUP BY
THREAD#;
MAX(SEQUENCE#)      THREAD#
-----
8                  1
```

3. Use a terminal window on `host03` connected as `oracle` with the environment variables set to `london`. Start SQL\*Plus and determine the last sequence number of the physical standby instance.

```
[oracle@host03 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Mon Jun 1 16:57:40
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> SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED_LOG GROUP BY
THREAD#;
MAX(SEQUENCE#)      THREAD#
-----
8                  1
```

4. Return to the terminal window of `host01`, 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;
SQL> SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED_LOG GROUP BY
THREAD#;
MAX(SEQUENCE#)      THREAD#
-----
9                  1
```

5. Return to the terminal window of `host03`, and verify that the 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#
-----
9                  1
```

6. Exit SQL\*Plus on both host01 and host03.

```
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.3.0.0.0
[oracle@host01 ~] $
```

```
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.3.0.0.0
[oracle@host03] $
```



# **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 12c Release 2 (12.2.0.1), password file changes done on the primary database are automatically propagated to standby databases.

### Tasks

1. Open a terminal window and use the SSH client to connect to host01 as the oracle OS user.

```
[oracle@gateway ~]$ ssh oracle@host01  
oracle@host01's password: <password>
```

2. Use the oraenv utility to set the environment variables for the boston instance.

```
[oracle@host01 ~]$ . oraenv  
ORACLE_SID = [oracle] ? boston  
The Oracle base has been set to /u01/app/oracle
```

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

```
[oracle@host01 ~]$ 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>
```

4. 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
```

5. Open a new terminal window and use the SSH client to connect to host03 as the oracle OS user.

```
[oracle@gateway ~]$ ssh oracle@host03
oracle@host03's password: <password>
```

6. Use the oraenv utility to set the environment variables for the london instance.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been set to /u01/app/oracle
```

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

```
[oracle@host03 ~]$ 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>
```

8. 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
```

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

```
SQL> alter database recover managed standby database cancel;
Database altered.
```

10. Return to the terminal connected to host01 and create a common user named C##DBA.

```
SQL> CREATE USER c##dba IDENTIFIED BY <password> container=all;
User created.
```

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

```
SQL> GRANT sysdba, create session TO c##dba container=all;
Grant succeeded.
```

12. 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          TRUE    FALSE
C##DBA      TRUE    FALSE
```

13. Return to the terminal session connected to host03 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
```

14. Start the Media Recovery Process (mrp0).

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

```
Database altered.
```

15. 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 host01 to see if the new entry can be added.

16. Return to the terminal session connected to host01 and change the password for C##DBA user.

```
SQL> ALTER USER c##dba IDENTIFIED BY Welcome_1;  
  
User altered.
```

17. Test the connection to the standby database (london) with the new password.

```
SQL> CONNECT c##dba/Welcome_1@london as sysdba  
Connected.
```

18. Drop the c##dba user.

```
SQL> connect / as sysdba  
Connected  
  
SQL> DROP USER c##dba CASCADE;  
  
User dropped.  
  
SQL>
```

19. Exit SQL\*Plus on host01 and host03 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.

### Tasks

1. Use the terminal session on `host01`. Create a directory for the new data files of `DEV2`.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [boston] ? boston
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@host01 ~]$ mkdir -p /u01/app/oracle/oradata/BOSTON/dev2
[oracle@host01 ~]$
```

2. 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@host01 ~]$ 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/BOSTON/dev2';

Pluggable database created.

SQL>
```

3. Check the open mode of `DEV2`.

```
SQL> show pdbs

  CON_ID CON_NAME           OPEN MODE  RESTRICTED
----- -----
    2 PDB$SEED            READ ONLY  NO
    3 DEV1                 READ WRITE NO
    4 DEV2                 MOUNTED

SQL>
```

4. Open DEV2 in READ WRITE mode.

```
SQL> alter pluggable database DEV2 open;  
Pluggable database altered.  
SQL>
```

5. Use the terminal session connected to host03. Connect as SYS to the london standby database.

```
[oracle@host03 ~]$ 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>
```

6. 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   string      *  
SQL>
```

7. Verify the PDBs in the london standby database.

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

```
SQL> show pdbs  
  
CON_ID CON_NAME          OPEN MODE  RESTRICTED  
-----  
    2  PDB$SEED           MOUNTED  
    3  DEV1                MOUNTED  
    4  DEV2                MOUNTED  
SQL>
```

8. 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.
```

9. Return to the terminal session on host01 connected to the boston database. Create a directory for DEV3.

```
SQL> !mkdir -p /u01/app/oracle/oradata/BOSTON/dev3  
  
SQL>
```

10. Create DEV3 from DEV1.

```
SQL> CREATE PLUGGABLE DATABASE dev3 FROM dev1  
CREATE_FILE_DEST='/u01/app/oracle/oradata/BOSTON/dev3';  
  
Pluggable database created.  
  
SQL>
```

11. Check the open mode of DEV3.

```
SQL> show pdbs  
  
CON_ID CON_NAME OPEN MODE RESTRICTED  
-----  
2 PDB$SEED READ ONLY NO  
3 DEV1 READ WRITE NO  
4 DEV2 READ WRITE NO  
5 DEV3 MOUNTED  
  
SQL>
```

12. Open DEV2 in READ WRITE mode.

```
SQL> alter pluggable database DEV3 open;  
  
Pluggable database altered.  
  
SQL>
```

13. Return to the terminal session on host03 connected to the london database. Verify the PDBs in the standby database.

```
SQL> show pdbs  
  
CON_ID CON_NAME OPEN MODE RESTRICTED  
-----  
2 PDB$SEED MOUNTED  
3 DEV1 MOUNTED  
4 DEV2 MOUNTED  
5 DEV3 MOUNTED
```

14. The DEV3 PDB is listed in the output of the previous step, but it doesn't mean 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	OUNTED	ENABLED
DEV1	OUNTED	ENABLED
DEV2	OUNTED	ENABLED
DEV3	OUNTED	DISABLED

```
SQL>
```

15. To clean up the environment, reset the `ENABLED_PDBS_ON_STANDBY` parameter.

```
SQL> ALTER SYSTEM SET enabled_pdbs_on_standby="*";
```

```
System altered.
```

16. Return to the terminal session connected to `host01` 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>
```

17. Exit SQL\*Plus on `host01` and `host03` leaving the terminal windows open for future practice.

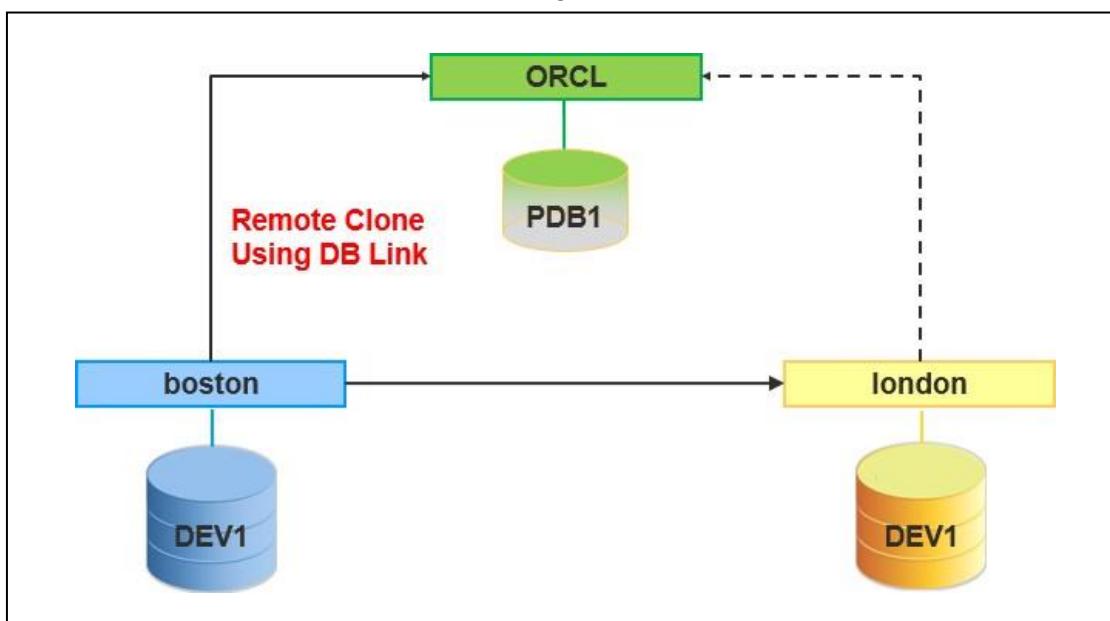
## Practice 4-3: Automating Instantiation of a PDB

### Overview

In this practice, you will test the usage of `STANDBY_PDB_SOURCE_FILE_DBLINK` to automate instantiation of a PDB in the `london` standby database when performing the remote PDB clone in the `boston` primary database. In Oracle Database 12c, 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

- `boston`: Primary database with a single PDB called `DEV1`
- `london`: 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.

```
[oracle@gateway ~]$ ssh oracle@em13c
oracle@em13c's password: <password>
Last login: Mon Jun  8 09:30:39 2020
[oracle@em13c ~]$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@em13c ~]$ 
[oracle@em13c ~]$ 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
----- -----
  2  PDB$SEED            READ ONLY   NO
  3  PDB1                 READ WRITE  NO
SQL>
```

2. 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>
```

3. Use the terminal window connected to host01 and create a database link in the boston primary database to the ORCL source database using the database link.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$ 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>
```

4. Test the database link from the boston primary database to the ORCL source database.

```
SQL> SELECT * FROM dual@clone_link;  
  
D  
-  
X  
  
SQL>
```

5. Use the terminal window connected to host03 and configure the london 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@host03 ~]$ . oraenv  
ORACLE_SID = [oracle] ? london  
The Oracle base has been set to /u01/app/oracle  
[oracle@host03 ~]$ 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> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT  
      FROM SESSION;
```

```
Database altered.
```

6. Test the database link from the `london` 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.
```

7. Return to the `ORCL` terminal session on `em13c` 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>
```

8. Return to the `boston` terminal session on `host01` 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/BOSTON';

Session altered.

SQL> CREATE PLUGGABLE DATABASE new_pdb1 FROM pdb1@clone_link;

Pluggable database created.

SQL>
```

9. Open a new terminal window connected to `host03` and review the alert log file.

```
[oracle@gateway ~]$ ssh host03
oracle@host03's password: <password>
Last login: Sun Jun 14 17:33:09 2020 from gateway.example.com
[oracle@host03 ~]$ tail -100
/u01/app/oracle/diag/rdbms/london/london/trace/alert_london.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
...
```

10. Return to the `ORCL` terminal session on `em13c`. 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.
```

11. Return to the `boston` terminal session on `host01` 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 (`boston`). 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      READ ONLY  ENABLED
DEV1          READ WRITE ENABLED
NEW_PDB1      READ WRITE ENABLED

SQL>
```

12. Return to terminal session connected to the `london` database on `host03` 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>
```

13. Return to the boston terminal session on host01 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>
```

14. Exit SQL\*Plus on em13c, host01, and host03 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 various offloadable workloads such as real-time query, DML/DDL on Global Temporary Tables, and read-mostly applications.

## 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 `host03` with the environment variables set for `london` appropriately. Make sure that the physical standby database and its `DEV1` PDB are in `READ ONLY` mode.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been set to /u01/app/oracle
[oracle@host03 ~]$
[oracle@host03 ~]$ 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
----- -----
  2 PDB$SEED            READ ONLY  NO
  3 DEV1                MOUNTED

SQL> alter pluggable database DEV1 open;

Pluggable database altered.
SQL>
```

2. **(Optional)** If the physical standby database is in the MOUNT state, stop the redo apply service and open the `london` 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>
```

3. 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>
```

4. 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
```

5. 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>
```

6. Leave the above window open. Open a terminal window (if not already open) logged in as oracle to host01 with the environment variables set for boston appropriately. Launch SQL\*Plus and switch session to the DEV1 PDB of the primary database. Query the HR.REGIONS table.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$ 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
```

7. 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.
```

8. Return to the SQL\*Plus session to the DEV1 PDB that is still open for the physical standby database on host03 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
```

9. Exit SQL\*Plus on `host03` of the physical standby database. It is recommended to keep the terminal session open with the environment variables set appropriately.
10. Exit SQL\*Plus on `host01`, 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 `london` 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 `host03` with the environment variables set for `london` appropriately. Check if the real time query is enabled in the `london` database as the system user.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been set to /u01/app/oracle
[oracle@host03 ~]$ 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;
```

OPEN_MODE	DATABASE_ROLE	DATAGUARD
<hr/>		
READ ONLY WITH APPLY PHYSICAL STANDBY DISABLED		

2. Attempt to create a Global Temporary Table (GTT) on Active Data Guard (read-only) instance.

```
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.

3. Use a terminal window logged in as `oracle` to host01 with the environment variables set for `boston` appropriately. Log in as `system` to the `boston` database.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$ 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

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

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

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

5. Return to the london terminal session on host03 and check the TEMP\_UNDO\_ENABLED parameter.

```
SQL> show parameter TEMP_UNDO_ENABLED
      NAME          TYPE        VALUE
-----+
temp_undo_enabled    boolean    FALSE
SQL>
```

6. 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.

7. Exit SQL\*Plus on host01 and host03, 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 `london` 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 `host03` with the environment variables set for `london` appropriately. Log in as the `SYSTEM` user.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been set to /u01/app/oracle
[oracle@host03 ~]$ 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>
```

2. 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
```

3. 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_
```

4. 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.

5. Insert rows in the PTT.

```
SQL> INSERT INTO ora$ptt_mine VALUES (1,'Work1');

1 row created.
```

6. Display data from the PTT.

```
SQL> SELECT * FROM ora$ptt_mine;

X  Y
-----
1  Work1
```

7. 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
```

8. Open another terminal window logged in as oracle to host03 with the environment variables set for london appropriately. Log in as the SYSTEM user.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been set to /u01/app/oracle
[oracle@host03 ~]$ 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>
```

9. 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
```

10. Return to the first SYSTEM window session on host03 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.

11. 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.
```

12. 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

13. Insert rows in the PTT.

```
SQL> INSERT INTO ora$ptt_mine2 VALUES (2, 'Work2');
```

```
1 row created.
```

14. Display data from the PTT.

```
SQL> SELECT * FROM ora$ptt_mine2;
```

X	Y
2	Work2

15. 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.

16. Exit SQL\*Plus on host01 and host03, 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 `host01` with the environment variables set for `boston` appropriately. Log in as the `SYS` user.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [boston] ? boston
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@host01 ~]$ 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>
```

2. 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           boolean    TRUE
SQL>
```

3. Use a terminal window logged in as `oracle` to host03 with the environment variables set for `london` appropriately. Log in as the `SYS` user.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [london] ? london
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@host03 ~]$ 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>
```

4. 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          boolean        TRUE
SQL>
```

5. Return to the terminal session connected to host01. Connect to the `DEV1` PDB for testing.

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

Session altered.

SQL> show con_name
```

```
CON_NAME
```

```
-----  
DEV1  
SQL>
```

6. 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
```

7. Return to the terminal session connected to host03. Connect to the DEV1 PDB.

```
SQL> alter session set container = DEV1;  
  
Session altered.  
  
SQL> show con_name  
  
CON_NAME  
-----  
DEV1
```

8. Display the data in the TEST01 table.

```
SQL> SELECT * FROM test01;  
  
ID NAME  
-----  
10 SEAN
```

9. 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:
//
    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
temporary
//
        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.

10. 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@host03 ~]$ sqlplus
sys/<password>@host03:1521/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.
```

11. Return to the terminal session connected to host01. Verify the result of the automatic redirection of DML feature in the boston primary database.

```
SQL> SELECT * FROM test01;
```

```
no rows selected
```

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

12. Exit SQL\*Plus on host01 and host03 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 (`bostonFS`) on `host02` that is in close proximity to the primary database.

### Tasks

1. Use a terminal window for `host01` logged in as `oracle` with the environment variables set to `boston`. Use SQL\*Plus to create a text-based initialization parameter file named `/tmp/initbostonFS.ora` that contains a copy of all the current parameters for the primary database.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$ 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/initbostonFS.ora' from spfile;

File created.
```

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

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

Database altered.

SQL> exit
```

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

```
[oracle@host01 ~]$ cp $ORACLE_HOME/dbs/orapwboston /tmp  
[oracle@host01 ~]$
```

4. Transfer the three files staged in the /tmp directory from host01 to host02 and place them into the /tmp directory on host02.

```
[oracle@host01 ~]$ cat /home/oracle/setup/copy_bostonFS.sh  
...  
scp /tmp/initbostonFS.ora oracle@host02:/tmp  
scp /tmp/bostonFS.ctl oracle@host02:/tmp  
scp /tmp/orapwboston oracle@host02:/tmp  
  
[oracle@host01 ~]$ /home/oracle/setup/copy_bostonFS.sh  
oracle@host02's password: <password>  
initbostonFS.ora          100% 1718      1.6MB/s  00:00  
oracle@host02's password: <password>  
bostonFS.ctl              100%    18MB   33.5MB/s  00:00  
oracle@host02's password: <password>  
orapwboston                100% 2560      2.6MB/s  00:00  
[oracle@host01 ~]$
```

5. Open a terminal window for host02 logged in as oracle with the environment variables set to bostonFS. Create the initial directories needed on the Far Sync server. These are the same directories that were created on the physical standby server host03 in practice 3-2.

```
[oracle@host02 ~]$ . oraenv  
ORACLE_SID = [oracle] ? bostonFS  
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/bostonFS/adump  
mkdir -p /u01/app/oracle/oradata/bostonFS  
mkdir -p /u01/app/oracle/oradata/bostonFS/pdbseed  
mkdir -p /u01/app/oracle/oradata/bostonFS/dev1  
mkdir -p /u01/app/oracle/fast_recovery_area/bostonFS  
  
[oracle@host02 ~]$ /home/oracle/setup/crdir_host02.sh  
[oracle@host02 ~]$
```

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

6. On host02, use the VI utility to edit the /tmp/initbostonFS.ora file. You will need to make changes to the initialization parameters to reflect the name change from boston to bostonFS. You will also need to make adjustments for control file name changes and directory name changes.

```
[oracle@host02 ~]$ vi /tmp/initbostonFS.ora
```

- a. Globally, search and replace all occurrences of boston with bostonFS.

```
:%s/boston/bostonFS/g
```

```
19 substitutions on 18 lines
```

- b. Globally, search and replace all occurrences of BOSTON with bostonFS.

```
:%s/BOSTON/bostonFS/g
```

```
2 substitutions on 1 lines
```

- c. Two of the substitutions from the global search and replace are incorrect and need to be back to their original value. Continue editing the file to locate and correct db\_name and log\_archive\_config 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='boston'
```

```
*.log_archive_config='dg_config=(boston,bostonFS, london, londonFS  
, london2)'
```

- d. 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=london ASYNC REOPEN=15  
valid_for=(STANDBY_LOGFILES, STANDBY_ROLE) db_unique_name=london'
```

- e. The original control files are named control01.ctl and control02.ctl. We will rename these bostonFS01.ctl and bostonFS02.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/bostonFS0/g
```

- f. Add the following new entries to the bottom of the file.

```
*.db_unique_name=bostonFS
```

```
*.fal_server=boston
```

```
*.log_file_name_convert='BOSTON' , 'bostonFS'
```

- g. Remove or comment out the LOCAL\_LISTENER entry if it exists with the value.

```
#*.local_listener='LISTENER_bostonFS'
```

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

```
:wq!
```

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

```
[oracle@host02 ~]$ cat /tmp/initbostonFS.ora
bostonFS._data_transfer_cache_size=0
bostonFS._db_cache_size=46137344
bostonFS._java_pool_size=12582912
bostonFS._large_pool_size=12582912
bostonFS._oracle_base='/u01/app/oracle'#ORACLE_BASE set from
environment
bostonFS._pga_aggregate_target=209715200
bostonFS._sga_target=310378496
bostonFS._shared_io_pool_size=4194304
bostonFS._shared_pool_size=226492416
bostonFS._streams_pool_size=0
*.audit_file_dest='/u01/app/oracle/admin/bostonFS/adump'
*.audit_trail='db'
*.compatible='12.1.0.0.0'
*.control_files='/u01/app/oracle/oradata/bostonFS/bostonFS01.ctl
','/u01/app/oracle/fast_recovery_area/bostonFS/bostonFS02.ctl'
*.db_block_size=8192
*.db_domain='example.com'
*.db_name='boston'
*.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=bostonFSXDB)'
*.enable_pluggable_database=true
#*.local_listener='LISTENER_bostonFS'
*.log_archive_config='dg_config=(boston,bostonFS,london, londonFS
,london2)'
*.log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST
valid_for=(ALL_LOGFILES,ALL_ROLES) db_unique_name=bostonFS'
*.log_archive_dest_2='SERVICE=london ASYNC REOPEN=15
valid_for=(STANDBY_LOGFILES,STANDBY_ROLE) db_unique_name=london'
*.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=bostonFS
```

```
*.fal_server=boston  
*.log_file_name_convert='BOSTON', 'bostonFS'
```

7. Copy the /tmp/bostonFS.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/bostonFS.ctl  
/u01/app/oracle/oradata/bostonFS/bostonFS01.ctl
```

```
[oracle@host02 ~]$ cp /tmp/bostonFS.ctl  
/u01/app/oracle/fast_recovery_area/bostonFS/bostonFS02.ctl
```

8. 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/orapwboston  
/u01/app/oracle/product/19.3.0/dbhome_1/dbs/orapwbostonFS
```

9. Verify that the environment variables are defined for bostonFS. 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=bostonFS  
ORAHOME=/u01/app/oracle/product/19.3.0/dbhome_1  
ORASID=oracle  
[oracle@host02 ~]$
```

10. 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/initbostonFS.ora';  
File created.
```

11. 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.
```

12. Use a terminal window on host01 logged in as oracle with the environment variables set to boston, the primary database. Launch SQL\*plus and examine the current value of log\_archive\_dest\_2.

```
[oracle@host01 ~]$ 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=london ASYNC REOPEN=15  
valid_for=(ONLINE_LOGFILES,
```

```
PRIMARY_ROLE)
db_unique_name=london
```

13. 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=bostonFS SYNC
REOPEN=15 valid_for=(ONLINE_LOGFILES,PRIMARY_ROLE)
db_unique_name=bostonFS' scope=both;
```

```
System altered.
```

14. 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
```

15. Use the previous SQL\*Plus session for host02 logged in as oracle with the environment variables set to bostonFS, 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
```

16. Open a terminal window on host03 logged in as oracle with the environment variables set to london. Launch SQL\*Plus and verify that the physical standby on host03 is receiving redo from the Far Sync on host02.

```
[oracle@host03]$ 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
```

17. Exit all SQL\*Plus sessions from host01, host02, and host03. Leave the terminal sessions open with the environment variables set for each system.

```
(host01) SQL> exit;
(host02) SQL> exit;
(host03) SQL> exit;
```

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

```
[oracle@host02 ~]$ ls -la
/u01/app/oracle/oradata/bostonFS/stdby*
-rw-r-----. 1 oracle oinstall 209715712 Jun  8 15:39
/u01/app/oracle/oradata/bostonFS/stdbyredo01.log
-rw-r-----. 1 oracle oinstall 209715712 Jun  8 15:37
/u01/app/oracle/oradata/bostonFS/stdbyredo02.log
-rw-r-----. 1 oracle oinstall 209715712 Jun  8 15:37
/u01/app/oracle/oradata/bostonFS/stdbyredo03.log
-rw-r-----. 1 oracle oinstall 209715712 Jun  8 15:38
/u01/app/oracle/oradata/bostonFS/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 (`londonFS`) 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 `london` instance on `host03` will become the primary database. It will need to forward redo to `londonFS` 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 `bostonFS`. Copy the three files staged in `/tmp` to `host04`.

```
[oracle@host02 ~]$ . oraenv
ORACLE_SID = [oracle] ? bostonFS
The Oracle base has been set to /u01/app/oracle
[oracle@host02 ~]$
[oracle@host02 ~]$ cat /home/oracle/setup/copy_londonFS.sh
...
scp /tmp/bostonFS.ctl oracle@host04:/tmp
scp /tmp/initbostonFS.ora oracle@host04:/tmp
scp /tmp/orapwboston oracle@host04:/tmp

[oracle@host02 ~]$ /home/oracle/setup/copy_londonFS.sh
/home/oracle/setup/copy_londonFS.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>
bostonFS.ctl          100%   18MB   52.9MB/s   00:00
oracle@host04's password: <password>
initbostonFS.ora      100% 1851     19.0KB/s   00:00
oracle@host04's password: <password>
orapwboston           100% 2560    239.8KB/s   00:00
[oracle@host02 ~]$
```

2. Open a new terminal window for host04 logged in as oracle with the environment variables set to londonFS. Create the initial directories needed on the Far Sync server. These are the same directories that were created on the physical standby server host03 in practice 3-2.

```
[oracle@host04 ~]$ . oraenv
ORACLE_SID = [oracle] ? londonFS
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/londonFS/adump
mkdir -p /u01/app/oracle/oradata/londonFS
mkdir -p /u01/app/oracle/oradata/londonFS/pdbseed
mkdir -p /u01/app/oracle/oradata/londonFS/dev1
mkdir -p /u01/app/oracle/fast_recovery_area/londonFS
[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 bostonFS and londonFS for readability. The last two letters are in uppercase.

3. Rename the /tmp/initbostonFS.ora file /tmp/initlondonFS.ora. You will need to make changes to the initialization parameters to reflect the name change from boston to london. You will also need to make adjustments for control file name changes and directory name changes.

```
[oracle@host04 ~]$ mv /tmp/initbostonFS.ora
/tmp/initlondonFS.ora
```

```
[oracle@host04 ~]$ vi /tmp/initlondonFS.ora
```

- a. Globally, search and replace all occurrences of boston with london

```
:%s/boston/london/g
26 substitutions on 22 lines
```

- b. 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='boston'
*.log_archive_config='dg_config=(boston,bostonFS,london,londonFS
, london2)'
*.log_file_name_convert='BOSTON,'londonFS'
```

- c. The entry for `log_archive_dest_2` is defined for a standby database role to ship redo to the `london` service. If `london` becomes the primary database, then the `londonFS` Far Sync should ship redo to `boston`, 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=boston ASYNC REOPEN=15
valid_for=(STANDBY_LOGFILES,STANDBY_ROLE) db_unique_name=boston'
```

- d. 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!
```

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

```
[oracle@host04 ~]$ cat /tmp/initlondonFS.ora
londonFS._data_transfer_cache_size=0
londonFS._db_cache_size=46137344
londonFS._java_pool_size=12582912
londonFS._large_pool_size=12582912
londonFS._oracle_base='/u01/app/oracle'#ORACLE_BASE set from
environment
londonFS._pga_aggregate_target=209715200
londonFS._sga_target=310378496
londonFS._shared_io_pool_size=4194304
londonFS._shared_pool_size=226492416
londonFS._streams_pool_size=0
*.audit_file_dest='/u01/app/oracle/admin/londonFS/adump'
*.audit_trail='db'
*.compatible='12.1.0.0.0'
*.control_files='/u01/app/oracle/oradata/londonFS/londonFS01.ctl
','/u01/app/oracle/fast_recovery_area/londonFS/londonFS02.ctl'
*.db_block_size=8192
*.db_domain='example.com'
*.db_name='boston'
*.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=londonFSXDB)'
*.enable_pluggable_database=true
#*.local_listener='LISTENER_londonFS'
*.log_archive_config='dg_config=(boston,bostonFS,london,londonFS,
london2)'
*.log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST
valid_for=(ALL_LOGFILES,ALL_ROLES) db_unique_name=londonFS'
```

```

*.log_archive_dest_2='SERVICE=boston ASYNC REOPEN=15
valid_for=(STANDBY_LOGFILES,STANDBY_ROLE) db_unique_name=boston'
*.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=londonFS
*.fal_server=london
*.log_file_name_convert='BOSTON', 'londonFS'

```

4. Copy the /tmp/bostonFS.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/bostonFS.ctl
/u01/app/oracle/oradata/londonFS/londonFS01.ctl

[oracle@host04 ~]$ cp /tmp/bostonFS.ctl
/u01/app/oracle/fast_recovery_area/londonFS/londonFS02.ctl

```

5. 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/orapwboston
/u01/app/oracle/product/19.3.0/dbhome_1/dbs/orapwlondonFS

```

6. Verify that the environment variables are defined for londonFS. 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=londonFS
ORAHOME=/u01/app/oracle/product/19.3.0/dbhome_1
ORASID=oracle

```

7. 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/initlondonFS.ora';

File created.
```

8. 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;
```

9. Use a terminal window on host03 logged in as oracle with the environment variables set to london, the physical standby database. Launch SQL\*plus and examine the current value of log\_archive\_dest\_2.

```
[oracle@host03 ~]$ 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

```

10. The `log_archive_dest_2` parameter of the `london` 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 `londonFS` Far Sync. Modify the `log_archive_dest_2` parameter so that it forwards redo accordingly.

```

SQL> alter system set log_archive_dest_2='SERVICE=londonFS SYNC
REOPEN=15 valid_for=(ONLINE_LOGFILES,PRIMARY_ROLE)
db_unique_name=londonFS' scope=both;

```

System altered.

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

11. Exit all SQL\*Plus sessions from `host01`, `host02`, `host03`, and `host04` if they are still open. Leave the terminal sessions open with the environment variables set for each system.

```

(host01) SQL> exit;
(host02) SQL> exit;
(host03) 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 `london` physical standby database to a snapshot standby database.

### Tasks

1. Use a terminal window logged in as `oracle` to `host03` with the environment variables set for `london` appropriately. Launch SQL\*Plus and verify that the current database role is physical standby.

```
[oracle@host03 ~]$ 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>
```

2. 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

3. 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
```

4. 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;

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
AUXILIARY DATAFILE COPY 0              0

8 rows selected.
```

5. 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
```

6. 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.

7. Display the current database role.

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

DATABASE\_ROLE

-----

SNAPSHOT STANDBY

8. 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

9. 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

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

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

OPEN_MODE
-----
MOUNTED
```

11. 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, number_of_files, percent_space_used from
v$recovery_area_usage;

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.
```

12. 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.

13. 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
```

14. 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.
```

15. In step 6, managed recovery was stopped for the snapshot database. Use a terminal window logged in as oracle to host01 to with the environment variables set for boston appropriately. Determine the last archived redo log for the primary database and perform a log switch. Exit SQL\*Plus when done.

```
[oracle@host01 ~]$ 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;
```

16. Return to the SQL\*Plus session on host03 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 host03. 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
```

2. 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.
```

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

```
SQL> alter database convert to physical standby;
Database altered.
```

4. 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;

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
AUXILIARY DATAFILE COPY 0                  0
```

5. 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.
```

6. Attempt to query the MISCL 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 miscl;
select * from miscl
      *
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
```

7. Return the session back to the container root. Shut down the physical standby to disable Active Data Guard. Start it 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.
```

8. 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 host03 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 host01 connected as oracle with the environment variables set to boston. 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.

```
[oracle@host01 ~]$ 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$ROOT      READ WRITE
          2 PDB$SEED      READ ONLY
          3 DEV1          READ WRITE
```

2. 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.

3. 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.

4. 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
```

5. 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@host01 ~] $
```

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

---

### Overview

In this practice, you will prepare host03 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 host03. Run the setup\_london2.sh script. The script will complete all preparation steps and create the london2 standby database.

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

```
[oracle@host03 ~]$ /home/oracle/setup/setup_london2.sh
#####
Task 1: Create Directories for london2
#####
```

Done

```
#####
Task 2: Create initlondon2.ora for london2
#####
```

Done

```
#####
Task 3: Create password file for london2
#####
```

Done

```
#####
Task 4: Start london2 NOMOUNT
#####
```

The Oracle base remains unchanged with value /u01/app/oracle

```
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.
```

```
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
```

```
Done
```

```
#####
# Task 5: Create london2 STANDBY DB
#####
```

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

```
Version 19.3.0.0.0
```

```
(c) 1982, 2019, Oracle and/or its affiliates. Allrights  
reserved.
```

```
connected to target database: BOSTON (DBID=2732274290)  
connected to auxiliary database: LONDON2 (not mounted)
```

```

RMAN> run {
2> allocate channel prmy1 type disk;
3> allocate auxiliary channel stby1 type disk;
4> duplicate target database for standby from active database
5> spfile
6> parameter_value_convert
('boston','london2','BOSTON','london2')
7>   set db_name='boston'
8>   set db_unique_name='london2'
9>   set db_file_name_convert='BOSTON','london2'
10>  set log_file_name_convert='BOSTON','london2'
11>  set fal_server='boston'
12>  set
log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST
valid_for=(ALL_LOGFILES,ALL_ROLES) db_unique_name=london2'
13> nofilenamecheck;
14> allocate auxiliary channel stby type disk;
15> sql channel stby "alter database recover managed standby
database disconnect"; }
16>
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/orapwlondon2' ;
  restore clone from service 'boston' spfile to

'/u01/app/oracle/product/19.3.0/dbhome_1/dbs/spfilelondon2.ora'
;

```

```

        sql clone "alter system set spfile=
'''/u01/app/oracle/product/19.3.0/dbhome_1/dbs/spfilelondon2.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 boston
channel stby1: restoring SPFILE
output file
name=/u01/app/oracle/product/19.3.0/dbhome_1/dbs/spfilelondon2.
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/spfilelondon2.ora
'''

contents of Memory Script:
{
    sql clone "alter system set audit_file_dest =
'''/u01/app/oracle/admin/london2/adump''' comment=
'''' scope=spfile";
    sql clone "alter system set control_files =
'''/u01/app/oracle/oradata/london2/control01.ctl''',
'''/u01/app/oracle/fast_recovery_area/london2/control02.ctl''' comment=
'''' scope=spfile";
    sql clone "alter system set dispatchers =
'''(PROTOCOL=TCP) (SERVICE=london2XDB)''' comment=
'''' scope=spfile";
    sql clone "alter system set local_listener =
'''LISTENER_london2''' comment=

```

```

      ''' scope=spfile";
      sql clone "alter system set db_name =
      ''boston'' comment=
      ''' scope=spfile";
      sql clone "alter system set db_unique_name =
      ''london2'' comment=
      ''' scope=spfile";
      sql clone "alter system set db_file_name_convert =
      ''BOSTON'', ''london2'' comment=
      ''' scope=spfile";
      sql clone "alter system set log_file_name_convert =
      ''BOSTON'', ''london2'' comment=
      ''' scope=spfile";
      sql clone "alter system set fal_server =
      ''boston'' 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=london2'''
      comment=
      ''' scope=spfile";
      shutdown clone immediate;
      startup clone nomount;
}
executing Memory Script

sql statement: alter system set audit_file_dest =
  ''/u01/app/oracle/admin/london2/adump'' comment= '''
scope=spfile

sql statement: alter system set control_files =
  ''/u01/app/oracle/oradata/london2/control01.ctl'',
  ''/u01/app/oracle/fast_recovery_area/london2/control02.ctl''
comment= ''' scope=spfile

sql statement: alter system set dispatchers =
  '(PROTOCOL=TCP) (SERVICE=london2XDB)' comment= '''
scope=spfile

```

```

sql statement: alter system set local_listener =
''LISTENER_london2'' comment= '''' scope=spfile

sql statement: alter system set db_name = ''boston'' comment=
'''' scope=spfile

sql statement: alter system set db_unique_name = ''london2''
comment= '''' scope=spfile

sql statement: alter system set db_file_name_convert =
''BOSTON'', ''london2'' comment= '''' scope=spfile

sql statement: alter system set log_file_name_convert =
''BOSTON'', ''london2'' comment= '''' scope=spfile

sql statement: alter system set fal_server = ''boston''
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=london2''
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 'boston' 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 boston
channel stby1: restoring control file
channel stby1: restore complete, elapsed time: 00:00:02
output file name=/u01/app/oracle/oradata/london2/control01.ctl
output file
name=/u01/app/oracle/fast_recovery_area/london2/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/london2/temp01.dbf";
    set newname for tempfile 2 to
"/u01/app/oracle/oradata/london2/pdbseed/temp012020-06-02_14-
25-16-052-PM.dbf";
    set newname for tempfile 3 to
"/u01/app/oracle/oradata/london2/dev1/temp01.dbf";
    switch clone tempfile all;
    set newname for datafile 1 to
"/u01/app/oracle/oradata/london2/system01.dbf";
    set newname for datafile 3 to
"/u01/app/oracle/oradata/london2/sysaux01.dbf";
    set newname for datafile 4 to

```

```

"/u01/app/oracle/oradata/london2/undotbs01.dbf";
      set newname for datafile 5 to
"/u01/app/oracle/oradata/london2/pdbseed/system01.dbf";
      set newname for datafile 6 to
"/u01/app/oracle/oradata/london2/pdbseed/sysaux01.dbf";
      set newname for datafile 7 to
"/u01/app/oracle/oradata/london2/users01.dbf";
      set newname for datafile 8 to
"/u01/app/oracle/oradata/london2/pdbseed/undotbs01.dbf";
      set newname for datafile 9 to
"/u01/app/oracle/oradata/london2/dev1/system01.dbf";
      set newname for datafile 10 to
"/u01/app/oracle/oradata/london2/dev1/sysaux01.dbf";
      set newname for datafile 11 to
"/u01/app/oracle/oradata/london2/dev1/undotbs01.dbf";
      set newname for datafile 12 to
"/u01/app/oracle/oradata/london2/dev1/users01.dbf";
      restore
      from nonparse   from service
'boston'    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/london2/temp01.dbf in control file
renamed tempfile 2 to
/u01/app/oracle/oradata/london2/pdbseed/temp012020-06-02_14-25-
16-052-PM.dbf in control file
renamed tempfile 3 to
/u01/app/oracle/oradata/london2/dev1/temp01.dbf in control file

```

```
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 boston
channel stby1: specifying datafile(s) to restore from backup
set
channel stby1: restoring datafile 00001 to
/u01/app/oracle/oradata/london2/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 boston
channel stby1: specifying datafile(s) to restore from backup
set
channel stby1: restoring datafile 00003 to
/u01/app/oracle/oradata/london2/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 boston
channel stby1: specifying datafile(s) to restore from backup
set
channel stby1: restoring datafile 00004 to
/u01/app/oracle/oradata/london2/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 boston
channel stby1: specifying datafile(s) to restore from backup
set
channel stby1: restoring datafile 00005 to
/u01/app/oracle/oradata/london2/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 boston
channel stby1: specifying datafile(s) to restore from backup
set
channel stby1: restoring datafile 00006 to
/u01/app/oracle/oradata/london2/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 boston
channel stby1: specifying datafile(s) to restore from backup
set
channel stby1: restoring datafile 00007 to
/u01/app/oracle/oradata/london2/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 boston
channel stby1: specifying datafile(s) to restore from backup
set
channel stby1: restoring datafile 00008 to
/u01/app/oracle/oradata/london2/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 boston
channel stby1: specifying datafile(s) to restore from backup
set
```

```
channel stby1: restoring datafile 00009 to
/u01/app/oracle/oradata/london2/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 boston
channel stby1: specifying datafile(s) to restore from backup
set
channel stby1: restoring datafile 00010 to
/u01/app/oracle/oradata/london2/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 boston
channel stby1: specifying datafile(s) to restore from backup
set
channel stby1: restoring datafile 00011 to
/u01/app/oracle/oradata/london2/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 boston
channel stby1: specifying datafile(s) to restore from backup
set
channel stby1: restoring datafile 00012 to
/u01/app/oracle/oradata/london2/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/london2/system01.dbf
datafile 3 switched to datafile copy
```

```
input datafile copy RECID=6 STAMP=1042203295 file
name=/u01/app/oracle/oradata/london2/sysaux01.dbf
datafile 4 switched to datafile copy
input datafile copy RECID=7 STAMP=1042203295 file
name=/u01/app/oracle/oradata/london2/undotbs01.dbf
datafile 5 switched to datafile copy
input datafile copy RECID=8 STAMP=1042203295 file
name=/u01/app/oracle/oradata/london2/pdbseed/system01.dbf
datafile 6 switched to datafile copy
input datafile copy RECID=9 STAMP=1042203295 file
name=/u01/app/oracle/oradata/london2/pdbseed/sysaux01.dbf
datafile 7 switched to datafile copy
input datafile copy RECID=10 STAMP=1042203295 file
name=/u01/app/oracle/oradata/london2/users01.dbf
datafile 8 switched to datafile copy
input datafile copy RECID=11 STAMP=1042203295 file
name=/u01/app/oracle/oradata/london2/pdbseed/undotbs01.dbf
datafile 9 switched to datafile copy
input datafile copy RECID=12 STAMP=1042203295 file
name=/u01/app/oracle/oradata/london2/dev1/system01.dbf
datafile 10 switched to datafile copy
input datafile copy RECID=13 STAMP=1042203295 file
name=/u01/app/oracle/oradata/london2/dev1/sysaux01.dbf
datafile 11 switched to datafile copy
input datafile copy RECID=14 STAMP=1042203295 file
name=/u01/app/oracle/oradata/london2/dev1/undotbs01.dbf
datafile 12 switched to datafile copy
input datafile copy RECID=15 STAMP=1042203295 file
name=/u01/app/oracle/oradata/london2/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
```

```
#####
#####
```

## Practice 8-3: Start Redo Transport and Verify Operation

---

### Overview

In this practice, you will start the redo transport from host01 to host03 for the new physical standby and verify operation.

### Tasks

1. Use a terminal window for host01 logged in as oracle with the environment variables set to boston and start redo transport by defining log\_archive\_dest\_3 pointing to the logical standby database.

```
[oracle@host01 ~]$ 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.

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=London2 SYNC
REOPEN=15 valid_for=(ONLINE_LOGFILES,PRIMARY_ROLE)
db_unique_name=London2' scope=both;
System altered.
```

**Note:** For this step, we are configuring redo transportation from the primary database to the logical standby database. Even though there is a far sync configured, we are not using the far sync at this moment. 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.

2. 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
```

3. Use a terminal window on host03 connected as oracle with the environment variables set to london2. Start SQL\*Plus and determine the last sequence number of the physical standby instance.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london2
The Oracle base has been set to /u01/app/oracle
[oracle@host03 ~]$ 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
```

4. Return to the terminal window of host01, 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
```

5. Return to the terminal window of host03, and verify that the london 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 'london2' to a logical standby database.

### Tasks

1. Use a terminal window on host03 connected as oracle with the environment variables set to london2. Stop redo apply on the london2 physical standby.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london2
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@host03 ~]$ 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 database recover managed standby database cancel;

Database altered.
```

2. Use a terminal window on host01 connected as oracle with the environment variables set to boston. 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.
```

3. Use a terminal window on host03 connected as oracle with the environment variables set to london2. Continue applying redo data to the physical standby until it is ready to convert to a logical standby database.

```
SQL> alter database recover to logical standby london2;

Database altered.
```

4. Increase the SGA size allocated to the logical standby database.

```
SQL> alter system set memory_max_target=840M scope=spfile;
System altered.

SQL> alter system set memory_target=840M scope=spfile;
System altered.
```

5. Shut down the logical standby database on host03, 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>
```

6. Display the LOG\_ARCHIVE\_DEST parameters on host03 that were copied from the primary database. Only entries that have values are displayed below.

```
SQL> show parameter log_archive_dest
NAME          TYPE        VALUE
-----
log_archive_dest_1 string      location=USE_DB_RECOVERY_FILE_
                                DEST valid_for = (ALL_LOGFILES,
                                ALL_ROLES) db_unique_name =
                                london2
log_archive_dest_2 string      SERVICE=bostonFS SYNC REOPEN=
                                15 valid_for=(ONLINE_LOGFILES,
                                PRIMARY_ROLE) db_unique_name=
                                bostonFS
```

7. Remove the LOG\_ARCHIVE\_DEST\_2 entry on host03 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.
```

8. Open the logical standby database.

```
SQL> alter database open resetlogs;
```

```
Database altered.
```

9. 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.
```

10. 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	READ WRITE
2	PDB\$SEED	READ ONLY
3	DEV1	READ WRITE

11. Exit SQL\*Plus sessions on all host machines. Leave the terminal session windows open with the environment variables set.

```
(host01) SQL> exit;
```

```
(host03) 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

1. Use a terminal window on host01 connected as oracle with the environment variables set to boston. Issue the "ID" command and verify that the dgdba operating system group is assigned to the oracle account.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01]$ 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.

2. Launch the DGMGRL utility and verify that you are able to connect as the SYSDG user with operating system authentication. (The oracle OS user is in the dgdba group.)

```
[oracle@host01 ~]$ 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: << Any Password will work >>
Connected to "boston"
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.

3. Attempt to make a remote connection to the physical standby database london as the SYSDG user. You must use the password that is in the password file. Exit DGMGRL.

```
DGMGRL> connect sysdg@london
Password: oracle_4U
ORA-01017: invalid username/password; logon denied

DGMGRL> exit
[oracle@host01 ~]$
```

**Note:** During the creation of the database, the option to use the same password for all administrative accounts was chosen. However, this applied only to the SYS and SYSTEM database accounts.

4. Use SQL\*Plus on host01 connected as SYSDBA to reset the SYSDG password and unlock the account. Exit SQL\*Plus.

```
[oracle@host01 ~]$ 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 <password>;
User altered.

SQL> alter user sysdg account unlock;
User altered.

SQL> grant sysdg to sysdg container=all;
Grant succeeded.

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

```
Version 19.3.0.0.0  
[oracle@host01 ~]$
```

5. In the current terminal window on host01, copy the modified password file to all other machines overwriting the password files that are already there. On host03, 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.

```
[oracle@host01 ~]$ cat /home/oracle/setup/scp_pwfile2.sh  
...  
scp $ORACLE_HOME/dbs/orapwboston  
host02:/u01/app/oracle/product/19.3.0/dbhome_1/dbs/orapwbostonFS  
scp $ORACLE_HOME/dbs/orapwboston  
host03:/u01/app/oracle/product/19.3.0/dbhome_1/dbs/orapwlondon  
scp $ORACLE_HOME/dbs/orapwboston  
host03:/u01/app/oracle/product/19.3.0/dbhome_1/dbs/orapwlondon2  
scp $ORACLE_HOME/dbs/orapwboston  
host04:/u01/app/oracle/product/19.3.0/dbhome_1/dbs/orapwlondonFS  
[oracle@host01 ~]$  
[oracle@host01 ~]$ /home/oracle/setup/scp_pwfile2.sh  
oracle@host02's password: <password>  
orapwboston          100% 2560      4.3MB/s  00:00  
oracle@host03's password: <password>  
orapwboston          100% 2560      3.9MB/s  00:00  
oracle@host03's password: <password>  
orapwboston          100% 2560      4.2MB/s  00:00  
oracle@host04's password: <password>  
orapwboston          100% 2560      5.5MB/s  00:00  
[oracle@host01 ~]$
```

**Note:** As of Oracle Database 12c Release 2 (12.2.0.1), 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.

6. Launch the DGMGRL utility on host01 and verify that you are now able to establish a remote connection as sysdg to the physical standby database. Exit DGMGRL when done.

```
[oracle@host01 ~]$ 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>@london
Connected to "london"
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 `host01` connected as `oracle` with the environment variables set to `boston`. 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@host01 ~]$ 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 log_archive_dest_2='' scope=both;

System altered.

SQL> alter system set log_archive_dest_3='' 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@host01 ~]$
```

2. Use a terminal window on host02 connected as oracle with the environment variables set to bostonFS. 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\*Plus.

```
[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 ~]$
```

3. Use a terminal window on `host03` connected as `oracle` with the environment variables set to `london`. 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@host03 ~]$ . oraenv
ORACLE_SID = [london2] ? london
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@host03 ~]$ 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.

SQL> alter system set log_archive_dest_2='' scope=both;

System altered.

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

System altered.
```

4. If you are using a separate window for the host03/london2 combination, then you can skip the part about resetting the environment variables. Otherwise, while still using the terminal window for host03, exit SQL\*Plus. Change the environment variables to the london2 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@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london2
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@host03 ~]$ 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> exit
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.3.0.0.0
[oracle@host03 ~]$
```

5. Use a terminal window on `host04` connected as `oracle` with the environment variables set to `londonFS`. Connect to Far Sync using SQL\*Plus and reset the `LOG_ARCHIVE_DEST_2` parameter because it is defined as network location. Start the Data Guard broker process for Far Sync. Exit SQL\*Plus.

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

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Jun 4 15:25:58
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@host04 ~]$
```

6. Use a terminal window on `host01` connected as `oracle` with the environment variables set to `boston`. Launch DGMGRL and attempt to show the configuration.

```
[oracle@host01 ~]$ 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>@boston
Connected to "boston"
Connected as SYSDG.
DGMGRL> show configuration
ORA-16532: Oracle Data Guard broker configuration does not exist

Configuration details cannot be determined by DGMGRL
DGMGRL>
```

7. Create the Data Guard broker configuration and then show the configuration.

```
DGMGRL> create configuration 'DRSolution' as primary database is
'boston' connect identifier is boston;
Configuration "DRSolution" created with primary database
"boston"
DGMGRL> show configuration

Configuration - DRSSolution

Protection Mode: MaxPerformance
Members:
boston - Primary database

Fast-Start Failover: Disabled

Configuration Status:
DISABLED

DGMGRL>
```

**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 (`host01`), 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.

8. Add the Far Sync bostonFS to the configuration and show the results.

```
DGMGRL> add far_sync 'bostonFS' as connect identifier is
bostonFS;
far sync instance "bostonFS" added
DGMGRL> show configuration

Configuration - DRSolution

Protection Mode: MaxPerformance
Members:
boston    - Primary database
bostonFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:
DISABLED
```

9. Add the physical standby database london to the configuration and show the results.

```
DGMGRL> add database 'london' as connect identifier is london;
Database "london" added
DGMGRL> show configuration

Configuration - DRSolution

Protection Mode: MaxPerformance
Members:
boston    - Primary database
bostonFS - Far sync instance
london    - Physical standby database

Fast-Start Failover: Disabled

Configuration Status:
DISABLED
```

10. Add the logical standby database london2 to the configuration and show the results.

```
DGMGRL> add database 'london2' as connect identifier is london2;
Database "london2" added
DGMGRL> show configuration

Configuration - DRSolution

Protection Mode: MaxPerformance
Members:
boston   - Primary database
bostonFS - Far sync instance
london   - Physical standby database
london2  - Logical standby database

Fast-Start Failover: Disabled

Configuration Status:
DISABLED

DGMGRL>
```

11. Add the Far Sync londonFS to the configuration and show the results.

```
DGMGRL> add far_sync 'londonFS' as connect identifier is
londonFS;
far sync instance "londonFS" added
DGMGRL> show configuration

Configuration - DRSolution

Protection Mode: MaxPerformance
Members:
boston   - Primary database
bostonFS - Far sync instance
london   - Physical standby database
london2  - Logical standby database
londonFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:
DISABLED
```

12. Enable the Data Guard broker configuration and show the results.

```
DGMGRL> enable configuration
Enabled.

DGMGRL> show configuration
Configuration - DRSolution

Protection Mode: MaxPerformance
Databases:
boston - Primary database
bostonFS - Far Sync (inactive)
london - Physical standby database
london2 - Logical standby database
londonFS - Far Sync (inactive)

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS
```

13. Define redo routing rules for the configuration and show the results. The current primary database `boston` should forward redo to the Far Sync `bostonFS` synchronously. The Far Sync `bostonFS` should forward redo to both the physical standby `london` and to the logical standby `london2` asynchronously. Additional redo routing rules should be created for role reversal. After role reversal, the primary database will be `london` and should forward redo to the Far Sync `londonFS` synchronously. The Far Sync `londonFS` should then forward redo to the physical standby `boston` and the logical standby `london2`.

```
DGMGRL> EDIT DATABASE 'boston' SET PROPERTY 'RedoRoutes' =
'(boston:bostonFS SYNC)';
Property "RedoRoutes" updated

DGMGRL> EDIT FAR_SYNC 'bostonFS' SET PROPERTY 'RedoRoutes' =
'(boston:london, london2 ASYNC)';
Property "RedoRoutes" updated

DGMGRL> EDIT DATABASE 'london' SET PROPERTY 'RedoRoutes' =
'(london:londonFS SYNC)';
Property "RedoRoutes" updated

DGMGRL> EDIT FAR_SYNC 'londonFS' SET PROPERTY 'RedoRoutes' =
'(london:boston, london2 ASYNC)';
Property "RedoRoutes" updated
```

```
DGMGRL> show configuration

Configuration - DRSolution

Protection Mode: MaxPerformance
Members:
boston - Primary database
bostonFS - Far sync instance
london - Physical standby database
london2 - Logical standby database

Members Not Receiving Redo:
londonFS - 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 london2 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 london2". 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 london 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

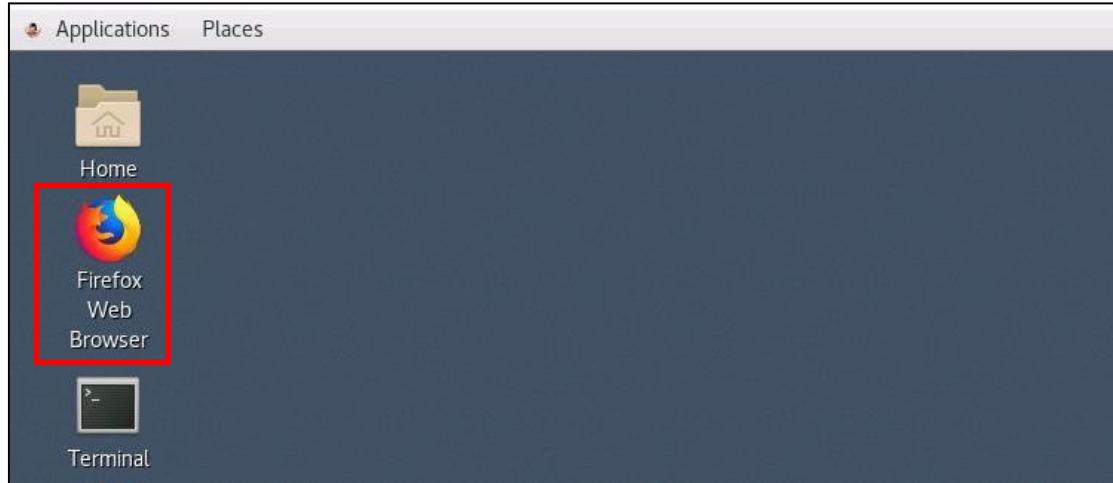
---

### 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.

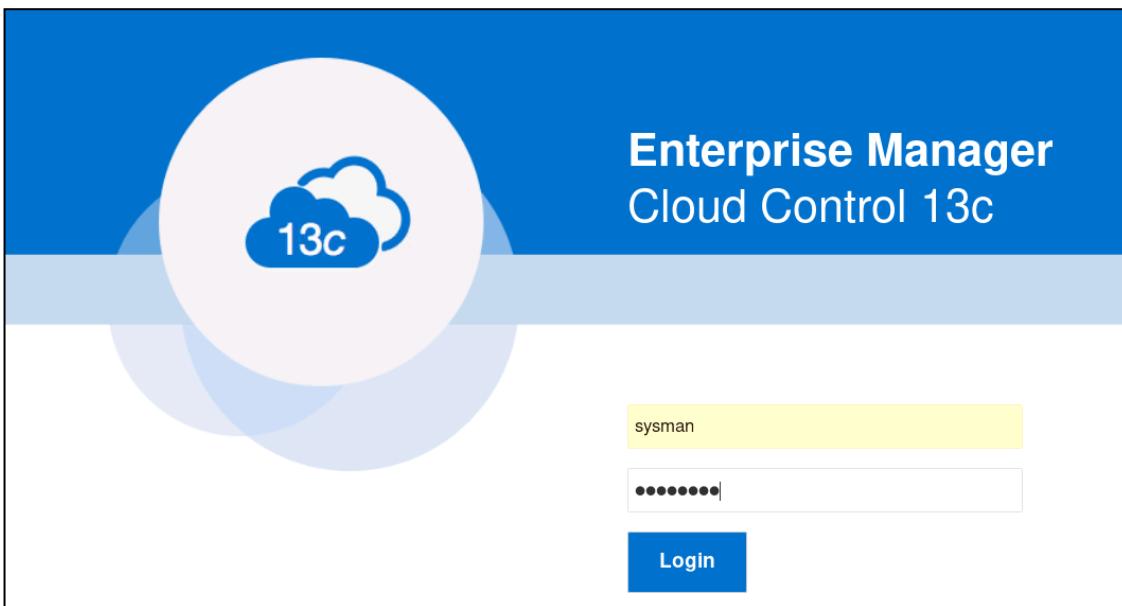
### Tasks

1. Using the Firefox Web Browser icon, start Firefox to access to Enterprise Manager 13c.



2. Enter the URL for Cloud Control. In the current setup, use  
<https://em13c.example.com:7803/em>

3. Log in to Enterprise Manager Cloud Control 13c. Log in to the application with **sysman** as the username and <password> as the password. The password is case-sensitive.



4. 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.

The image shows the "Databases" page of Oracle Enterprise Manager Cloud Control 13c. The top navigation bar includes links for "All Targets", "Groups", "Systems", "Services", "Hosts", "Databases" (which is highlighted), and "Middleware". A search bar at the top allows filtering by "Name". Below the search bar is a toolbar with buttons for "View", "Add", "Remove", and "Configure". The main content area displays a table of database targets. The columns are "Name" and "Type". The data in the table is as follows:

Name	Type	Incidents	Average Compliance Score
boston.example.com	Database Instance : Container	0	N/A
ORCL	Database Instance : Container	0	N/A
rcatcdb	Database Instance : Container	0	N/A

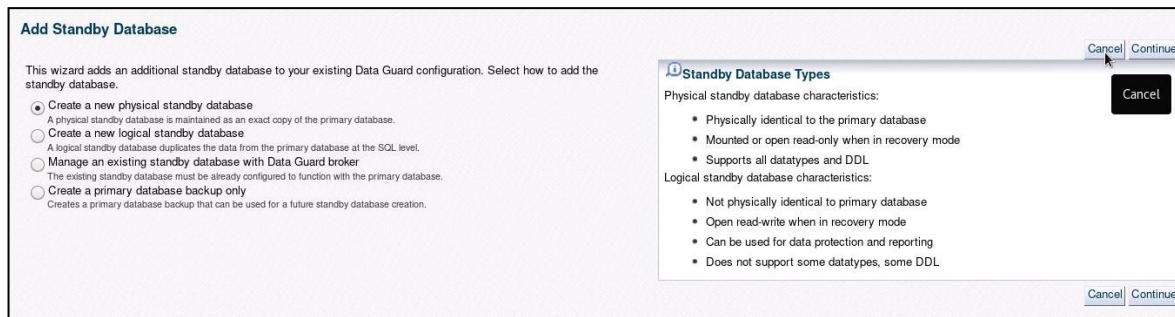
5. On the Databases page, click the link for the boston.example.com target.

Name	Type	Status	Target Version	Incidents	Average Compliance Score	Member Status Summary
boston.example.com	Database Instance : Container, Primary	↑	19.3.0.0.0	0 0 0	N/A	0 1 0 0
ORCL	Database Instance : Container	↑	19.3.0.0.0	0 0 0	N/A	0 1 0 0
rcatodbs	Database Instance : Container	↑	19.3.0.0.0	0 0 0	N/A	0 2 0 0

6. On the boston.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.



- 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\_BOSTON\_SYS

Database Login	
* Database:	boston.example.com (Container Database)
Credential:	<input type="radio"/> Preferred <input type="radio"/> Named <input checked="" type="radio"/> New
* Username:	sys
* Password:	*****
Role:	SYSDBA
<input checked="" type="checkbox"/> Save As: NC_BOSTON_SYS <input type="checkbox"/> Set As Preferred Credentials	
<input type="button" value="Login"/> <input type="button" value="Cancel"/>	

8. The Data Guard home page, you will see the data guard configuration that you created in practice 10-2.

**Data Guard**  
Page Refreshed June 10, 2020 3:33:13 AM UTC

**Overview**

Data Guard Status	✓ Normal
Protection Mode	Maximum Performance
Fast-Start Failover	Disabled

**Primary Database**

Name	boston
Host	host01.example.com
Data Guard Status	✓ Normal
Current Log	43
Properties	Edit

**Standby Databases**

Select	Name	Host	Data Guard Status	Role	Redo Source	Real-time Query	Last Received Log
<input checked="" type="radio"/>	bostonFS	host02.example.com	✓ Normal	Far Sync	boston	N/A	Not available
<input type="radio"/>	london	host03.example.com	✓ Normal	Physical Standby	bostonFS	Disabled	42
<input type="radio"/>	london2	host03.example.com	✓ Normal	Logical Standby	bostonFS	N/A	42
<input type="radio"/>	londonFS	host04.example.com	✓ Normal	Far Sync	Unknown	N/A	Not available

**Standby Database Progress Summary**

Transport lag is the time difference between the last update on the primary database and the last received database and the last applied redo on the standby database.

0.0 0.2 0.4 0.6 0.8 1.0 1.2  
seconds

Apply Lag Transport Lag

9. Review the **Overview** and **Primary Database** sections. It shows the status of the Data Guard configuration and the primary database.

**Overview**

Data Guard Status	✓ Normal
Protection Mode	Maximum Performance
Fast-Start Failover	Disabled

**Primary Database**

Name	boston.example.com
Host	host01.example.com
Data Guard Status	✓ Normal
Current Log	43
Properties	Edit

10. Review the **Standby Databases** section. It shows the members of the Data Guard configuration and role/status of each member.

Standby Databases					
	<a href="#">Edit</a>	<a href="#">Remove</a>	<a href="#">Switchover</a>	<a href="#">Failover</a>	<a href="#">Convert</a>
Select	Name	Host	Data Guard Status	Role	
<input checked="" type="radio"/>	bostonFS	host02.example.com	Normal	Far Sync	
<input type="radio"/>	london.example.com	host03.example.com	Normal	Physical Standby	
<input type="radio"/>	london2.example.com	host03.example.com	Normal	Logical Standby	
<input type="radio"/>	londonFS	host04.example.com	Normal	Far Sync	

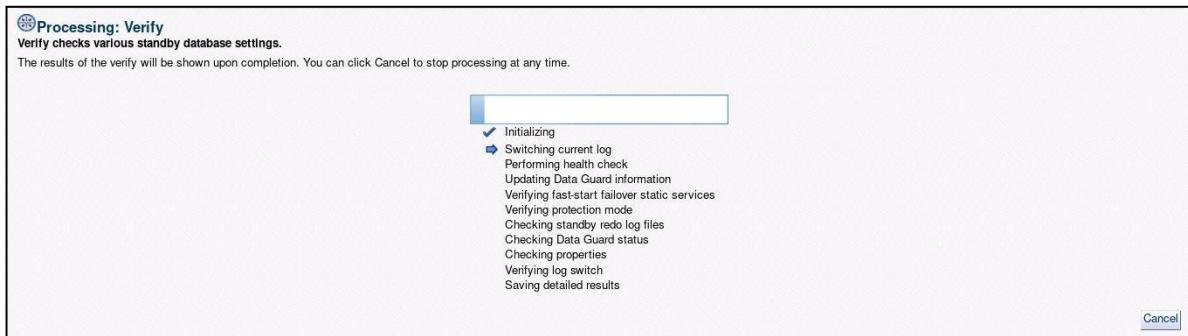
11. Continue to review the **Standby Databases** section. It shows the Redo Source of each member in the Data Guard configuration and additional information. In this section, you can also Add Far Sync or Add Standby Database.

					<a href="#">Add Far Sync</a>	<a href="#">Add Standby Database</a>
Redo Source	Real-time Query	Last Received Log	Last Applied Log	Estimated Failover Time		
boston	N/A	Not available	Not available	Not available		
bostonFS		42	42	Not available		
bostonFS	N/A	42	42	Not available		
Unknown	N/A	Not available	Not available	Not available		

12. Review the **Performance** and **Additional Administration** sections. It shows the additional links for the Data Guard performance.

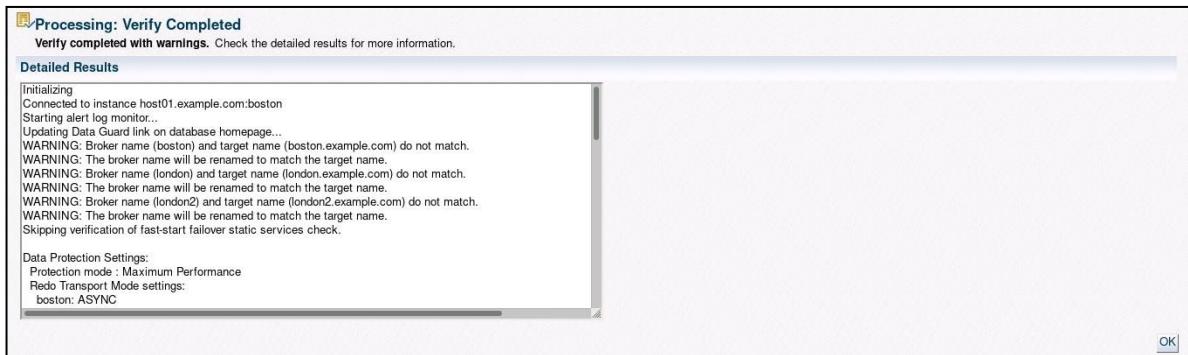
<b>Performance</b>	<b>Additional Administration</b>
<a href="#">Data Guard Performance</a>	<a href="#">Verify Configuration</a>
<a href="#">Log File Details</a>	<a href="#">Remove Data Guard Configuration</a>

13. 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.

14. 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 host01 connected as oracle with the environment variables set to boston. Launch DGMGRL connecting as the SYSDG user with operating system authentication.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$ 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>@boston
Connected to "boston"
Connected as SYSDG.
DGMGRL>
```

2. Use the SHOW CONFIGURATION VERBOSE command to display the current values for the CommunicationTimeout property and the OperationTimeout property.

```
DGMGRL> show configuration verbose

Configuration - DRSSolution

Protection Mode: MaxPerformance
Members:
boston   - Primary database
bostonFS - Far sync instance
london   - Physical standby database
london2  - Logical standby database
```

```

Members Not Receiving Redo:
londonFS - Far sync instance

Properties:
  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   = 'boston_CFG'

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS

DGMGRL>

```

3. Modify the **CommunicationTimeout** property and set it to a value of 300. Verify the result.

```

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'

```

```
FastStartFailoverAutoReinstate = 'TRUE'  
...
```

**Note:** This is not normally needed but it helps with labs running in the Virtual Machine architecture.

4. 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'  
...
```

5. Use the SHOW DATABASE command for the physical standby database and determine the current transport lag, apply lag, and apply rate.

```
DGMGRL> show database london  
  
Database - london  
  
Role:                  PHYSICAL STANDBY  
Intended State:        APPLY-ON  
Transport Lag:         0 seconds (computed 0 seconds ago)  
Apply Lag:             0 seconds (computed 0 seconds ago)  
Average Apply Rate:   17.00 KByte/s  
Real Time Query:      OFF  
Instance(s):  
    london  
  
Database Status:  
SUCCESS  
  
DGMGRL>
```

6. Stop redo apply on the physical standby database to force an apply rate lag to occur.

```
DGMGRL> edit database london set state = 'APPLY-OFF';
Succeeded.
```

7. 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.
```

8. Use the SHOW DATABASE command for the physical standby database and display the current apply lag rate.

```
DGMGRL> show database london
Database - london

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):
    london

Database Status:
SUCCESS
```

9. Display the standby receive queue for the physical standby database.

```
DGMGRL> show database london 'RecvQEntries';
STANDBY_RECEIVE_QUEUE
      STATUS      RESETLOGS_ID      THREAD
LOG_SEQ      TIME_GENERATED      TIME_COMPLETED
FIRST_CHANGE#          NEXT_CHANGE#      SIZE (KBs)
      PARTIALLY_APPLIED      1042035828      1
63 06/04/2020 18:55:17 06/04/2020 19:03:47
3030103          3032604      4151

DGMGRL>
```

10. (Optional) With the navigation techniques learned in practice 10-3, navigate to the `boston.example.com` database home page. On the `boston.example.com` database home page, click **Data Guard Administration** in the Availability menu.

The screenshot shows the Oracle Enterprise Manager Cloud Control 13c interface for the `boston.example.com` database. The main menu bar includes Availability, Security, Schema, and Administration. The Availability dropdown is open, showing options like MAA Advisor, Backup & Recovery, Add Standby Database, Data Guard Administration (which is highlighted with a cursor), Data Guard Performance, and Verify Data Guard Configuration. The top right corner displays the host name as `host01.example.com`, the refresh status as "Auto Refresh Off", and the page refresh time as "Page Refreshed Jun 10, 2020 1:29:22 PM GMT". The main dashboard shows version 19.3.0.0.0, 0 days, 23 hrs up time, 94.4% availability for last 7 days, and N/A last backup. A graph at the bottom shows Active Sessions over time from 1:18 AM to 2:18 AM, with a significant peak around 1:38 AM.

(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.

This screenshot is similar to the previous one, but the "Add Standby Database..." link under the Availability menu is now selected, indicated by a cursor. The rest of the interface remains the same, including the main dashboard and the activity session graph.

(Optional) If the Add Standby Database link shows the Add Standby Database page, click **Cancel** to navigate to the Data Guard home page.

The screenshot shows the "Add Standby Database" wizard. The first step, "Create a new physical standby database", is selected. It lists several options: "Create a new physical standby database", "Create a new logical standby database", "Manage an existing standby database with Data Guard broker", and "Create a primary database backup only". To the right, the "Standby Database Types" section is displayed, divided into "Physical standby database characteristics" (which includes being physically identical to the primary database, mounted or open read-only in recovery mode, and supporting all datatypes and DDL) and "Logical standby database characteristics" (which includes being not physically identical to the primary database, open read-write in recovery mode, used for data protection and reporting, and not supporting some datatypes and DDL). At the bottom right of the wizard are "Cancel" and "Continue" buttons.

11. (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\_BOSTON\_SYS

The screenshot shows the 'Database Login' page for 'boston.example.com (Container Database)'. The 'Credential' section has 'New' selected. The 'Username' field contains 'sys', the 'Password' field contains masked text, and the 'Role' dropdown is set to 'SYSDBA'. A checked checkbox labeled 'Save As' is followed by the value 'NC\_BOSTON\_SYS'. There is also an unchecked checkbox for 'Set As Preferred Credentials'. At the bottom are 'Login' and 'Cancel' buttons.

12. (Optional) On the Data Guard home page, you will see the status of redo apply service.

Standby Databases				
	Edit   Remove   Switchover   Failover   Convert			
Select	Name	Host	Data Guard Status	Role
<input checked="" type="radio"/>	bostonFS	host02.example.com	✓ Normal	Far Sync
<input type="radio"/>	london.example.com	host03.example.com	✓ Normal, Redo Apply Off	Physical Standby
<input type="radio"/>	london2.example.com	host03.example.com	✓ Normal	Logical Standby
<input type="radio"/>	londonFS	host04.example.com	✓ Normal	Far Sync

13. (Optional) Click the Log File Details link in the Performance section.

14. (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.

london.example.com		london2.example.com								
Log	Status	ResetLogs ID #	First Change # (SCN)	Last Change # (SCN)	Size (KB)	Redo Apply Services	Status	Apply Delay (minutes)	Time Generated	Time Completed
44	Partially Applied	1042163641	3741406	3743852	1036	On	Normal	0	Jun 10, 2020 2:11:45 PM	Jun 10, 2020 2:22:41 PM
44	Not Received	1042163641	3741406	3743852	1036	On	Normal	0	Jun 10, 2020 2:11:45 PM	Jun 10, 2020 2:22:41 PM

All logs have been received and applied.

15. 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 london set state = 'APPLY-ON';
Succeeded.
```

```
DGMGRL> show database london
```

```
Database - london
```

```
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):
london
```

```
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 host01 connected as oracle with the environment variables set to boston. 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@host01 ~]$ 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@host01 ~]$
```

2. 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@host01]$ cd
/u01/app/oracle/diag/rdbms/boston/boston/trace
```

3. Verify that the previous commands that changed the state of redo apply and connection timeout were recorded in the Data Guard broker log file. The broker log file is named drc<db\_unique\_name>.log.

```
[oracle@host01 trace]$ grep CommunicationTimeout drcboston.log
EDIT CONFIGURATION SET PROPERTY CommunicationTimeout = 300
Property "CommunicationTimeout" value set to "300" seconds
EDIT CONFIGURATION SET PROPERTY CommunicationTimeout = 300
completed successfully

[oracle@host01 trace]$ grep APPLY drcboston.log
EDIT DATABASE london SET STATE = APPLY-OFF
EDIT DATABASE london SET STATE = APPLY-OFF completed
successfully
EDIT DATABASE london SET STATE = APPLY-ON
EDIT DATABASE london SET STATE = APPLY-ON completed successfully
[oracle@host01 trace]$
```

4. 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@host01 trace]$ ls -alt | more
total 17812
-rw-r-----. 1 oracle oinstall 3130 Jun 4 19:09
boston_m000_7209.trc
-rw-r-----. 1 oracle oinstall 1015 Jun 4 19:09
boston_m000_7209.trm
-rw-r-----. 1 oracle oinstall 12215 Jun 4 19:09
boston_m001_5486.trc
-rw-r-----. 1 oracle oinstall 1602 Jun 4 19:09
boston_m001_5486.trm
-rw-r-----. 1 oracle oinstall 14706 Jun 4 19:09
boston_mmon_5423.trc
-rw-r-----. 1 oracle oinstall 1594 Jun 4 19:09
boston_mmon_5423.trm
-rw-r-----. 1 oracle oinstall 11207 Jun 4 19:08
boston_m004_5819.trc
...
<Ctrl + C>
[oracle@host01 trace]$
```

5. 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@host01 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@host01 trace]$
```

6. 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@host01 trace]$ ls -alt | more
total 18152
-rw-r-----. 1 oracle oinstall 181546 Jun 4 19:11
boston_lgwr_5343.trc
-rw-r-----. 1 oracle oinstall    2818 Jun 4 19:11
boston_lgwr_5343.trm
-rw-r-----. 1 oracle oinstall 165194 Jun 4 19:11
boston_nss2_5502.trc
-rw-r-----. 1 oracle oinstall    2675 Jun 4 19:11
boston_nss2_5502.trm
```

```
...
<Ctrl + C>
[oracle@host01 trace]$
```

7. 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 bostonFS far sync destination, verify that communication occurred to this destination in the trace file using the "grep bostonFS <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@host01 trace]$ grep bostonFS boston_nss2_5343.trc
*rfsnam: /u01/app/oracle/oradata/bostonFS/stdbyredo02.log
[oracle@host01 trace]$
```

8. 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@host01 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@host01 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 host01 as the oracle user and set the environment.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [boston] ? boston
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@host01 ~]$
```

2. Using DGMGRL, connect to the boston database.

```
[oracle@host01 ~]$ 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>@boston
Connected to "boston"
Connected as SYSDG.
DGMGRL>
```

3. 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>
```

4. Use the VALIDATE command to perform a comparison of SPFILE entries between the boston primary database and the london standby database.

```
DGMGRL> VALIDATE DATABASE london SPFILE;  
Connecting to "boston".  
Connected to "boston"  
  
Connecting to "london".  
Connected to "london"  
  
Parameter settings with different values:  
  
audit_file_dest:  
boston (PRIMARY) : /u01/app/oracle/admin/boston/adump  
london          : /u01/app/oracle/admin/london/adump  
  
dispatchers:  
boston (PRIMARY) : (PROTOCOL=TCP) (SERVICE=bostonXDB)  
london          : (PROTOCOL=TCP) (SERVICE=londonXDB)  
  
enabled_PDBs_on_standby:  
boston (PRIMARY) : NOT SPECIFIED  
london          : *  
  
log_archive_trace:  
boston (PRIMARY) : 0  
london          : 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 london SPFILE.

5. Display a brief summary of the bostonFS and londonFS Far Sync instances where the london database becomes the new primary database.

```
DGMGRL> VALIDATE FAR_SYNC bostonFS WHEN PRIMARY IS london;

Member Role:          Far Sync Instance
When Primary Is:    london
Redo Destinations:   (none)

Thread #  Online Redo Log Groups  Standby Redo Log Groups Status
              london                  bostonfs
1           3                           2           Insufficient SRLs

DGMGRL> VALIDATE FAR_SYNC londonFS WHEN PRIMARY IS london;

Member Role:          Far Sync Instance
When Primary Is:    london
Redo Destinations:   boston
                      london2

Thread #  Online Redo Log Groups  Standby Redo Log Groups Status
              london                  londonfs
1           3                           2           Insufficient SRLs

DGMGRL>
```

6. Validate network configuration for the london database.

```
DGMGRL> VALIDATE NETWORK CONFIGURATION FOR london;
Connecting to instance "london" on database "london" ...
Connected to "london"
Checking connectivity from instance "london" on database "london"
to instance "boston" on database "boston"...
Succeeded.
Checking connectivity from instance "london" on database "london"
to instance "bostonFS" on database "bostonFS"...
Succeeded.
Checking connectivity from instance "london" on database "london"
to instance "london2" on database "london2"...
Succeeded.
Checking connectivity from instance "london" on database "london"
to instance "londonFS" on database "londonFS"...
Succeeded.
Connecting to instance "boston" on database "boston" ...
Connected to "boston"
Checking connectivity from instance "boston" on database "boston"
to instance "london" on database "london"...
Succeeded.
Connecting to instance "bostonFS" on database "bostonFS" ...
Connected to "bostonFS"
```

```

Checking connectivity from instance "bostonFS" on database
"bostonFS" to instance "london" on database "london"...
Succeeded.
Connecting to instance "london2" on database "london2" ...
Connected to "london2"
Checking connectivity from instance "london2" on database "london2"
to instance "london" on database "london"...
Succeeded.
Connecting to instance "londonFS" on database "londonFS" ...
Connected to "londonFS"
Checking connectivity from instance "londonFS" on database
"londonFS" to instance "london" on database "london"...
Succeeded.

Oracle Clusterware is not configured on database "london".
Connecting to database "london" using static connect identifier
"(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=host03.example.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=london_DGMGRL.example.com)(INSTANCE_NAME=london)(SERVER=DEDICATED)(STATIC_SERVICE=TRUE)))" ...
Succeeded.
The static connect identifier allows for a connection to database
"london".

DGMGRL>

```

## 7. Validate network configuration for all members.

```

DGMGRL> VALIDATE NETWORK CONFIGURATION FOR all;
Connecting to instance "boston" on database "boston" ...
Connected to "boston"
Checking connectivity from instance "boston" on database "boston"
to instance "bostonFS" on database "bostonFS"...
Succeeded.
Checking connectivity from instance "boston" on database "boston"
to instance "london" on database "london"...
Succeeded.
Checking connectivity from instance "boston" on database "boston"
to instance "london2" on database "london2"...
Succeeded.
Checking connectivity from instance "boston" on database "boston"
to instance "londonFS" on database "londonFS"...
Succeeded.
Connecting to instance "bostonFS" on database "bostonFS" ...
Connected to "bostonFS"
Checking connectivity from instance "bostonFS" on database
"bostonFS" to instance "boston" on database "boston"...
Succeeded.
Checking connectivity from instance "bostonFS" on database
"bostonFS" to instance "london" on database "london"...
Succeeded.
Checking connectivity from instance "bostonFS" on database
"bostonFS" to instance "london2" on database "london2"...
Succeeded.

```

```

Checking connectivity from instance "bostonFS" on database
"bostonFS" to instance "londonFS" on database "londonFS"...
Succeeded.
Connecting to instance "london" on database "london" ...
Connected to "london"
Checking connectivity from instance "london" on database "london"
to instance "boston" on database "boston"...
Succeeded.
Checking connectivity from instance "london" on database "london"
to instance "bostonFS" on database "bostonFS"...
Succeeded.
Checking connectivity from instance "london" on database "london"
to instance "london2" on database "london2"...
Succeeded.
Checking connectivity from instance "london" on database "london"
to instance "londonFS" on database "londonFS"...
Succeeded.
Connecting to instance "london2" on database "london2" ...
Connected to "london2"
Checking connectivity from instance "london2" on database "london2"
to instance "boston" on database "boston"...
Succeeded.
Checking connectivity from instance "london2" on database "london2"
to instance "bostonFS" on database "bostonFS"...
Succeeded.
Checking connectivity from instance "london2" on database "london2"
to instance "london" on database "london"...
Succeeded.
Checking connectivity from instance "london2" on database "london2"
to instance "londonFS" on database "londonFS"...
Succeeded.
Connecting to instance "londonFS" on database "londonFS" ...
Connected to "londonFS"
Checking connectivity from instance "londonFS" on database
"londonFS" to instance "boston" on database "boston"...
Succeeded.
Checking connectivity from instance "londonFS" on database
"londonFS" to instance "bostonFS" on database "bostonFS"...
Succeeded.
Checking connectivity from instance "londonFS" on database
"londonFS" to instance "london" on database "london"...
Succeeded.
Checking connectivity from instance "londonFS" on database
"londonFS" to instance "london2" on database "london2"...
Succeeded.

Oracle Clusterware is not configured on database "boston".
Connecting to database "boston" using static connect identifier
"(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=host01.example.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=boston_DGMGRL.example.com)(INSTANCE_NAME=boston)(SERVER=DEDICATED)(STATIC_SERVICE=TRUE)))" ...
Succeeded.

```

```
The static connect identifier allows for a connection to database "boston".
```

```
Oracle Clusterware is not configured on database "london".
Connecting to database "london" using static connect identifier
"(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=host03.example.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=london_DGMGRL.example.com)(INSTANCE_NAME=london)(SERVER=DEDICATED)(STATIC_SERVICE=TRUE)))" ...
Succeeded.
```

```
The static connect identifier allows for a connection to database "london".
```

```
Oracle Clusterware is not configured on database "london2".
Connecting to database "london2" using static connect identifier
"(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=host03.example.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=london2_DGMGRL.example.com)(INSTANCE_NAME=london2)(SERVER=DEDICATED)(STATIC_SERVICE=TRUE)))" ...
Succeeded.
```

```
The static connect identifier allows for a connection to database "london2".
```

```
DGMGRL>
```

#### 8. Validate the static connect identifier of the boston database.

```
DGMGRL> VALIDATE STATIC CONNECT IDENTIFIER FOR boston;
Oracle Clusterware is not configured on database "boston".
Connecting to database "boston" using static connect identifier
"(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=host01.example.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=boston_DGMGRL.example.com)(INSTANCE_NAME=boston)(SERVER=DEDICATED)(STATIC_SERVICE=TRUE)))" ...
Succeeded.
```

```
The static connect identifier allows for a connection to database "boston".
```

```
DGMGRL>
```

#### 9. Validate the static connect identifier for all databases.

```
DGMGRL> VALIDATE STATIC CONNECT IDENTIFIER FOR all;
Oracle Clusterware is not configured on database "boston".
Connecting to database "boston" using static connect identifier
"(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=host01.example.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=boston_DGMGRL.example.com)(INSTANCE_NAME=boston)(SERVER=DEDICATED)(STATIC_SERVICE=TRUE)))" ...
Succeeded.
```

```
The static connect identifier allows for a connection to database "boston".
```

```
Oracle Clusterware is not configured on database "london".
Connecting to database "london" using static connect identifier
"(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=host03.example.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=london_DGMGRL.example.com)(INSTANCE_NAME=london)(SERVER=DEDICATED)(STATIC_SERVICE=TRUE)))" ...
Succeeded.
```

```
T=1521)) (CONNECT_DATA=(SERVICE_NAME=london_DGMGRL.example.com) (INSTANCE_NAME=london) (SERVER=DEDICATED) (STATIC_SERVICE=TRUE)))" ...  
Succeeded.
```

The static connect identifier allows for a connection to database "london".

```
Oracle Clusterware is not configured on database "london2".  
Connecting to database "london2" using static connect identifier  
"(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=host03.example.com)(PORT=1521)) (CONNECT_DATA=(SERVICE_NAME=london2_DGMGRL.example.com) (INSTANCE_NAME=london2) (SERVER=DEDICATED) (STATIC_SERVICE=TRUE)))" ...  
Succeeded.
```

The static connect identifier allows for a connection to database "london2".

```
DGMGRL>
```

10. Exit DGMGRL on host01 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 host01 connected as oracle with the environment variables set to boston. Launch the DGMGRL utility and connect as the sysdg user with operating system authentication.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01]$ 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>@boston
Connected to "boston"
Connected as SYSDG.
```

2. 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:
boston   - Primary database
bostonFS - Far sync instance
london   - Physical standby database
london2  - Logical standby database

Members Not Receiving Redo:
```

```

londonFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS (status updated 58 seconds ago)

DGMGRL>

```

- Using DGMGRL, determine the current LogXptMode for the far sync instance and the physical standby database.

```

DGMGRL> show far_sync 'bostonFS' 'LogXptMode';
LogXptMode = 'ASYNC';

DGMGRL> show database london 'LogXptMode';
LogXptMode = 'ASYNC';
DGMGRL> exit
[oracle@host01 ~]$

```

**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 'bostonFS' would require it to be enclosed by single quotes. The second edit command does not use any single quotes for illustration.

- Optional:** 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 'bostonFS' set property 'LogXptMode' =
'ASYNC';
Property "LogXptMode" updated

DGMGRL> edit database london set property LogXptMode = ASYNC;
Property "logxptmode" updated

DGMGRL> exit
[oracle@host01 ~]$

```

5. 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@host01 ~]$ 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="bostonfs", SYNC
                                         AFFIRM delay=0 optional
                                         compression=disable
                                         max_failure=0
                                         max_connections=1 reopen=300
                                         db_unique_name="bostonFS"
                                         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@host01 ~]$
```

6. Launch the DGMGRL utility and connect as the SYSDG user.

```
[oracle@host01 ~]$ 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>@boston  
Connected to "boston"  
Connected as SYSDG.  
DGMGRL>
```

7. Display the value for the RedoRoutes property of the primary database.

```
DGMGRL> show database 'boston' 'RedoRoutes';  
RedoRoutes = '(boston:bostonFS SYNC)'
```

**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'.

8. Modify the 'RedoRoutes' property for the boston primary database and set it to the 'ASYNC' redo transport mode.

```
DGMGRL> edit database boston set property 'RedoRoutes' =  
'(boston:bostonFS ASYNC)';  
Property "RedoRoutes" updated
```

9. 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.
```

10. Modify the RedoRoutes property for the boston primary database and set it to the 'FASTSYNC' redo transport mode.

```
DGMGRL> edit database boston set property 'RedoRoutes' =  
'(boston:bostonFS FASTSYNC)';  
Property "RedoRoutes" updated
```

11. Change the configuration mode to maximum availability and verify the results.

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

Configuration - DRSSolution

**Protection Mode: MaxAvailability**

Members:

boston - Primary database  
bostonFS - Far sync instance  
london - Physical standby database  
london2 - Logical standby database

Members Not Receiving Redo:

londonFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:

SUCCESS (status updated 60 seconds ago)

DGMGRL>

12. Use a terminal window on host03 connected as oracle with the environment variables set to london. Connect to the physical standby database using SQL\*Plus and perform a shutdown abort.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base remains unchanged with value /u01/app/oracle

[oracle@host03 ~]$ 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>
```

13. Return to the DGMGRL session running on host01 and display the configuration.

```
DGMGRL> show configuration  
  
Configuration - DRSSolution  
  
Protection Mode: MaxAvailability  
Members:  
    boston   - Primary database  
    bostonFS - Far sync instance  
        Error: ORA-16778: redo transport error for one or more  
        members  
  
    london   - Physical standby database  
        Error: ORA-1034: ORACLE not available  
  
    london2  - Logical standby database  
  
Members Not Receiving Redo:  
    londonFS - Far sync instance  
  
Fast-Start Failover: Disabled  
  
Configuration Status:  
ERROR    (status updated 55 seconds ago)  
  
DGMGRL>
```

14. Return to the SQL\*Plus session on host03 connected as oracle with the environment variables set to london. 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	

15. Return to the DGMGRL session running on host01 and display the configuration.

```
DGMGRL> show configuration
```

```
Configuration - DRSSolution
```

```
Protection Mode: MaxAvailability
```

```
Members:
```

```
boston - Primary database
```

```
bostonFS - Far sync instance
```

```
Error: ORA-16778: redo transport error for one or more members
```

```
london - Physical standby database
```

```
Warning: ORA-16809: multiple warnings detected for the member
```

```
london2 - Logical standby database
```

```
Members Not Receiving Redo:
```

```
londonFS - 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.

16. Restart redo apply for the physical standby database. Perform a log switch on the primary database and verify the configuration.

```
DGMGRL> edit database london set state = 'APPLY-ON';
Succeeded.

DGMGRL> SQL "alter system switch logfile";
Succeeded.

DGMGRL> show configuration

Configuration - DRSolution

Protection Mode: MaxAvailability
Members:
boston      - Primary database
bostonFS   - Far sync instance
london      - Physical standby database
london2     - Logical standby database

Members Not Receiving Redo:
londonFS   - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS      (status updated 56 seconds ago)

DGMGRL>
```

17. 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 london commands until the lag clears. Repeat these commands as needed.

```
DGMGRL> show database london

Database - london

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):
london
```

## 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 `boston` primary database and set it to the `SYNC` redo transport mode. Enable the maximum protection mode for the Data Guard configuration.

```
DGMGRL> edit database boston set property 'RedoRoutes' =
  '(boston:bostonFS 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

Failed.
```

**Note:** The maximum protection mode is not supported by far sync.

2. Modify the redo transport routes so that the primary database ships redo to both the `bostonFS` far sync and the `london` physical standby database directly using the `SYNC` property. Modify the `bostonFS` far sync so that it only ships redo to the `london2` logical standby database using the `ASYNC` property. We are temporarily bypassing the far sync between the primary database and the physical standby database, but leaving the far sync in place for the logical standby database. This is for lab illustration only and would not be practical otherwise.

```
DGMGRL> edit far_sync 'bostonFS' set property 'RedoRoutes' =
  '(boston:london2 ASYNC)';
Property "RedoRoutes" updated

DGMGRL> edit database boston set property 'RedoRoutes' =
  '(boston:bostonFS, london SYNC)';
Property "RedoRoutes" updated
```

3. Enable the maximum protection mode for the Data Guard configuration and display the resulting configuration. Exit DGMGRL.

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

DGMGRL> show configuration

Configuration - DRSSolution

Protection Mode: MaxProtection
Members:
boston - Primary database
bostonFS - Far sync instance
london2 - Logical standby database
london - Physical standby database

Members Not Receiving Redo:
londonFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS (status updated 57 seconds ago)

DGMGRL> exit
[oracle@host01 ~]$
```

4. In the same terminal window on `host01`, connect to the primary database using SQL\*Plus and switch the session to the `DEV1` pluggable database. Leave this window open.

```
[oracle@host01 ~]$ 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>
```

5. Use a terminal window on `host03` connected as `oracle` with the environment variables set to `london`. Connect to the physical standby database using SQL\*Plus and perform a shutdown abort.

```
[oracle@host03 ~]$ . oraenv  
ORACLE_SID = [oracle] ? london  
The Oracle base has been set to /u01/app/oracle  
[oracle@host03 ~]$ 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>
```

6. Return to the SQL\*Plus session on `host01` with the session set to the `DEV1` pluggable database. Display the current data for the `HR.REGION` 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@host01 ~]$
```

**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.

7. In the same terminal window on `host01`, connect to the primary database using SQL\*Plus and attempt to restart the instance. Exit the terminated session.

```
[oracle@host01 ~]$ 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@host01 ~]$
```

**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.

8. Return to the SQL\*Plus session on `host03` 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
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

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.3.0.0.0
[oracle@host03 ~]$
```

9. Return to the terminal window on `host01` connected as `oracle` with the environment variables set to `boston`. Use SQL\*Plus to open the primary database. Verify that the pluggable database is open. If not, then open it. Exit SQL\*Plus.

```
[oracle@host01 ~]$ 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
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
----- -----
        2 PDB$SEED          READ ONLY   NO
        3 DEV1                READ WRITE

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.3.0.0.0
[oracle@host01 ~]$
```

10. In the same terminal window on **host01**, launch the DGMGRL utility and connect as the SYSDG user. Display the Data Guard configuration. Wait until the ORA-\* warning messages are cleared.

```
[oracle@host01 ~]$ 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>@boston
Connected to "boston"
Connected as SYSDG.
DGMGRL> show configuration

Configuration - DRSolution

Protection Mode: MaxProtection
Members:
boston      - Primary database
bostonFS   - Far sync instance
london2     - Logical standby database
london      - Physical standby database

Members Not Receiving Redo:
londonFS   - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS    (status updated 45 seconds ago)

DGMGRL>
```

11. Change the Data Guard protection mode to maximum performance.

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

12. Correct the redo routing rules such that the `boston` primary database forwards redo to only the far sync using the `FASTSYNC` attribute. Adjust the far sync to forward redo to both the `london` physical standby database and the `london2` logical standby database using the `ASYNC` attribute.

```
DGMGRL> edit database boston set property 'RedoRoutes' =
  '(boston:bostonFS SYNC)';
Property "RedoRoutes" updated

DGMGRL> edit far_sync 'bostonFS' set property 'RedoRoutes' =
  '(boston:london,london2 ASYNC)';
Property "RedoRoutes" updated
```

13. Restart Redo Apply on the physical standby database and perform a log switch on the primary database.

```
DGMGRL> edit database london set state = 'APPLY-ON';
Succeeded.
DGMGRL> SQL "alter system switch logfile";
Succeeded.
```

14. Display the resulting configuration.

```
DGMGRL> show configuration

Configuration - DRSSolution

Protection Mode: MaxPerformance
Members:
  boston      - Primary database
  bostonFS    - Far sync instance
  london      - Physical standby database
  london2     - Logical standby database

Members Not Receiving Redo:
  londonFS   - Far sync instance

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:

15. 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 london' commands until the lag clears. Repeat these commands as needed.

```
DGMGRL> show database london
Database - london

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):
    london

Database Status:
SUCCESS
```

16. 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 AWR in a Data Guard environment. You will also generate ASH In-Memory and ADDM report for an Active Data Guard instance.

## 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 `host01` with the environment variables set for `boston` appropriately. Launch SQL\*Plus on your primary database and determine if redo compression is enabled by querying `V$ARCHIVE_DEST`.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$ 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      DISABLE
LOG_ARCHIVE_DEST_2      DISABLE
LOG_ARCHIVE_DEST_3      DISABLE
LOG_ARCHIVE_DEST_4      DISABLE
LOG_ARCHIVE_DEST_5      DISABLE
LOG_ARCHIVE_DEST_6      DISABLE
LOG_ARCHIVE_DEST_7      DISABLE
LOG_ARCHIVE_DEST_8      DISABLE
LOG_ARCHIVE_DEST_9      DISABLE
LOG_ARCHIVE_DEST_10     DISABLE
```

LOG_ARCHIVE_DEST_11	DISABLE
DEST_NAME	COMPRES
-----	
LOG_ARCHIVE_DEST_12	DISABLE
LOG_ARCHIVE_DEST_13	DISABLE
LOG_ARCHIVE_DEST_14	DISABLE
LOG_ARCHIVE_DEST_15	DISABLE
LOG_ARCHIVE_DEST_16	DISABLE
LOG_ARCHIVE_DEST_17	DISABLE
LOG_ARCHIVE_DEST_18	DISABLE
LOG_ARCHIVE_DEST_19	DISABLE
LOG_ARCHIVE_DEST_20	DISABLE
LOG_ARCHIVE_DEST_21	DISABLE
LOG_ARCHIVE_DEST_22	DISABLE
DEST_NAME	COMPRES
-----	
LOG_ARCHIVE_DEST_23	DISABLE
LOG_ARCHIVE_DEST_24	DISABLE
LOG_ARCHIVE_DEST_25	DISABLE
LOG_ARCHIVE_DEST_26	DISABLE
LOG_ARCHIVE_DEST_27	DISABLE
LOG_ARCHIVE_DEST_28	DISABLE
LOG_ARCHIVE_DEST_29	DISABLE
LOG_ARCHIVE_DEST_30	DISABLE
LOG_ARCHIVE_DEST_31	DISABLE
31 rows selected.	
SQL>	

2. Use a terminal window logged in as `oracle` to `host02` with the environment variables set for `bostonFS` appropriately. Launch DGMGRL and connect to the primary database.

[oracle@host02 ~]\$ . oraenv
ORACLE_SID = [oracle] ? <b>bostonFS</b>
The Oracle base has been set to /u01/app/oracle
[oracle@host02 ~]\$ <b>dgmgrl</b>
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>@boston
```

```
Connected to "boston"
```

```
Connected as SYSDG.
```

```
DGMGRL>
```

3. Enable redo compression by setting the `RedoCompression` property for your far sync. Exit DGMGRL.

```
DGMGRL> edit far_sync 'bostonFS' set property
```

```
'RedoCompression'='ENABLE';
```

```
Property "RedoCompression" updated
```

```
DGMGRL> exit
```

```
[oracle@host02 ~]$
```

4. Return to your SQL\*Plus session on host01 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	DISABLE
--------------------	---------

LOG_ARCHIVE_DEST_2	ENABLE
--------------------	--------

LOG_ARCHIVE_DEST_3	DISABLE
--------------------	---------

...

31 rows selected.

```
SQL> show parameter log_archive_dest_2
```

NAME	TYPE	VALUE
------	------	-------

log_archive_dest_2	string	service="bostonfs", SYNC AFFIRM delay=0 optional <b>compression=enable</b> max_failure=0 max_connections=1 reopen=300 db_unique_name="bostonFS" net_timeout=30, valid_for=(online_logfile
--------------------	--------	--

```
        ,all_roles)

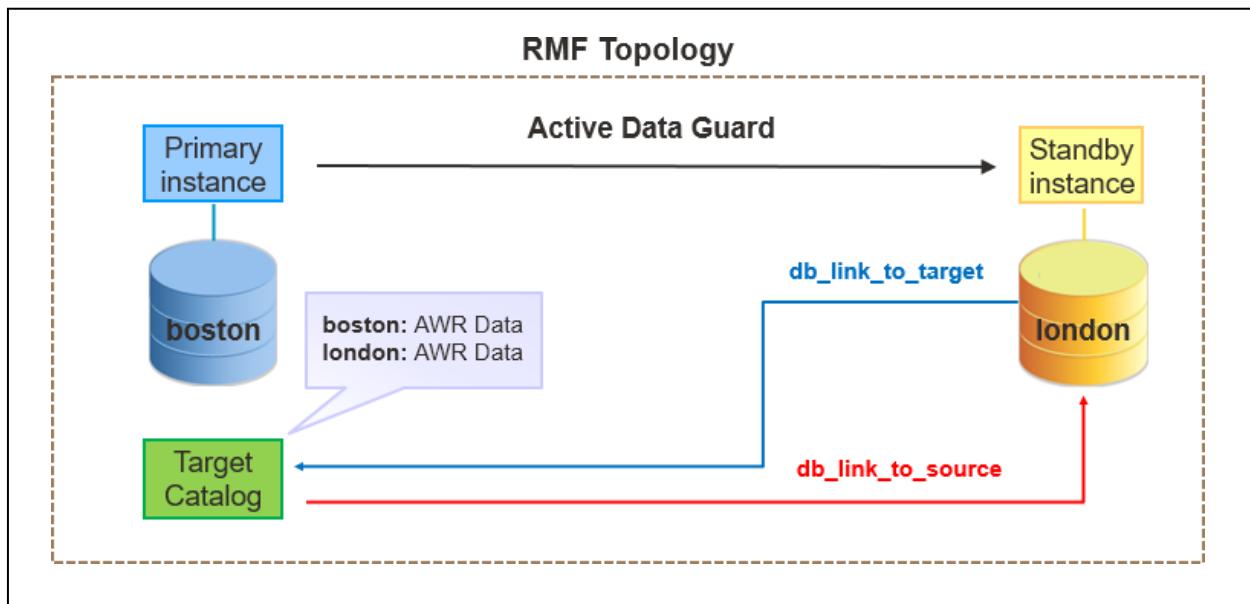
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.3.0.0.0
[oracle@host01 ~]$
```

## Practice 13-2: Generating AWR Report for an Active Data Guard Instance

### Overview

In this practice, you will configure Standby Automatic Workload Repository (AWR). Starting with Oracle Database 12.2, the Automated Workload Repository (AWR) can be configured to take snapshots of Active Data Guard standby databases. This feature enables analyzing any performance-related issues for ADG standby databases.

The AWR framework is enhanced to support capture of remote snapshots from any generic database, including Active Data Guard (ADG) databases. This framework is called the Remote Management Framework (RMF).



### Tasks

1. Use a terminal window logged in as `oracle` to `host03` with the environment variables set for `london` appropriately. Launch SQL\*Plus on your `london` standby database and confirm the open mode.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been set to /u01/app/oracle
[oracle@host03 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 13:00:24
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> **select open\_mode, database\_role from v\$database;**

OPEN_MODE	DATABASE_ROLE
MOUNTED	PHYSICAL STANDBY

SQL>

2. If the london physical standby database is not in Active Data Guard with the real-time query mode, configure your physical standby database properly. This feature requires Active Data Guard with the real-time query mode.

SQL> **alter database open;**

Database altered.

SQL> **alter pluggable database dev1 open;**

Pluggable database altered.

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> select open_mode, database_role from v$database;

OPEN_MODE          DATABASE_ROLE
-----
READ ONLY WITH APPLY PHYSICAL STANDBY

SQL>
```

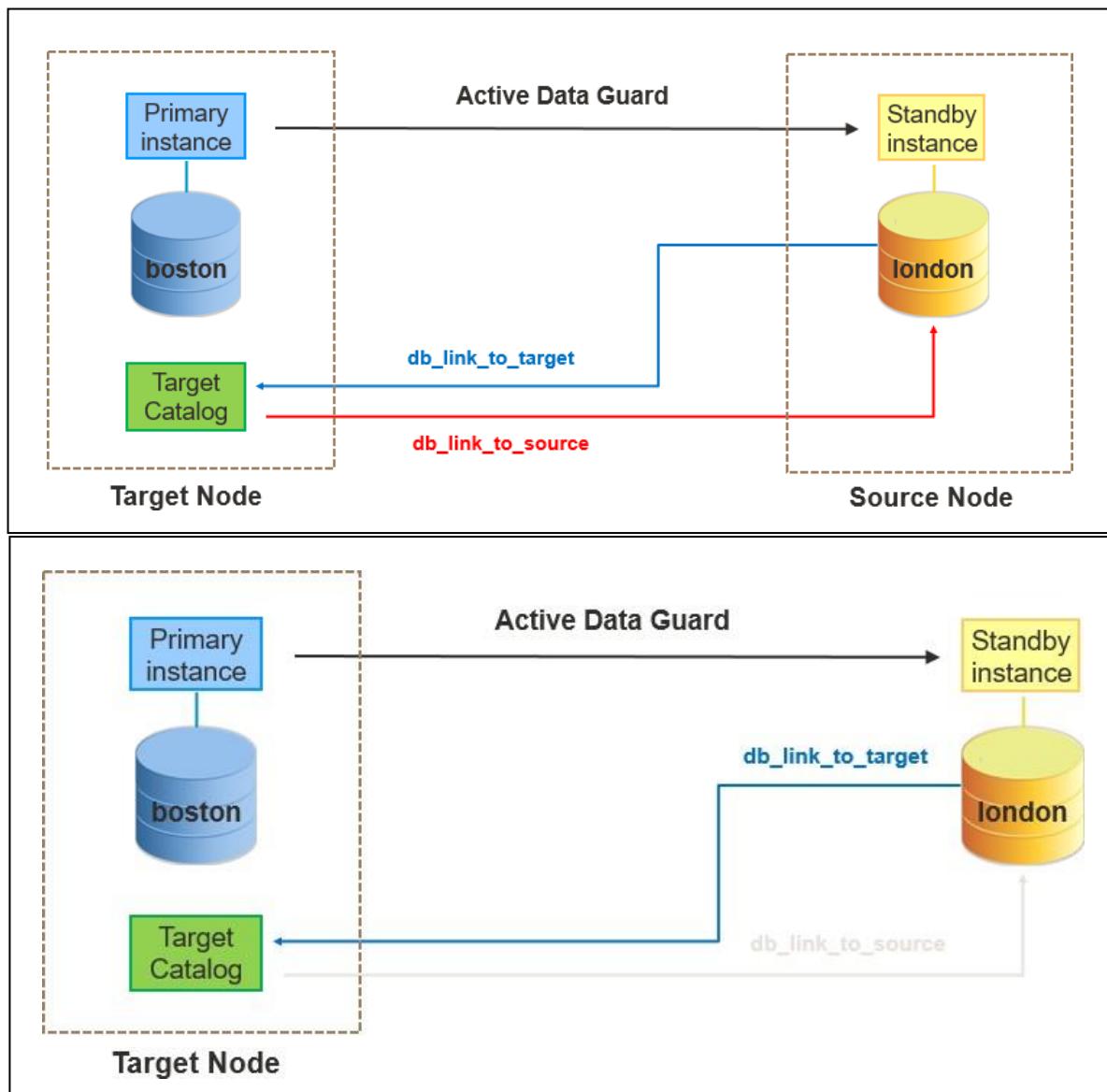
3. From this point, you will review a series of tasks performed for Standby AWR configuration. Use a terminal window logged in as `oracle` to `host01` with the environment variables set for `boston` appropriately. Launch SQL\*Plus on your `boston` standby database and confirm the open mode.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [boston] ? boston
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@host01 ~]$
```

4. Review the `AWR_s1.txt` file. It shows several parameter settings used for this practice.

```
[oracle@host01 ~]$ cat /home/oracle/setup/AWR_s1.txt
ptns=boston
stns=london
sysumf_passwd=oracle_4U
primary_name=boston
standby_name=london
stby2pri_link=london_to_boston
pri2stby_link=boston_to_london
topology_alias=Topology_Demo
[oracle@host01 ~]$
```

5. Review the `AWR_s2.txt` file. In this practice, we configure the AWR repository in the primary database. Therefore, it is called the target node.



```
[oracle@host01 ~]$ cat /home/oracle/setup/AWR_s2.txt
echo "1. Create node"
echo "1.1 Create target node name at $ptns"

sqlplus /nolog <<EOF
spool crumf_pnode.log
connect sys/oracle_4U@$ptns as sysdba
alter system set "_umf_remote_enabled"=TRUE scope=BOTH;
alter user sys\$umf account unlock identified by $sysumf_passwd;
```

```

drop database link $stby2pri_link;
create database link $stby2pri_link connect to sys\$umf
identified by $sysumf_passwd using '$ptns';
exec dbms_umf.unconfigure_node;
exec dbms_umf.configure_node('$primary_name');
select dbms_umf.get_node_name_local as node_name from dual;
EOF
grep ORA- crumf_pnode.log | grep -v ORA-02024
if [ $? -eq 0 ]
then
echo "Error: Failed to create target node at $ptns."
echo "Please check crumf_pnode.log for more detail"
exit 2
fi
[oracle@host01 ~]$

```

6. Review the AWR\_s3.txt file. This step shows how to make the london physical standby database as a source node of the AWR data.

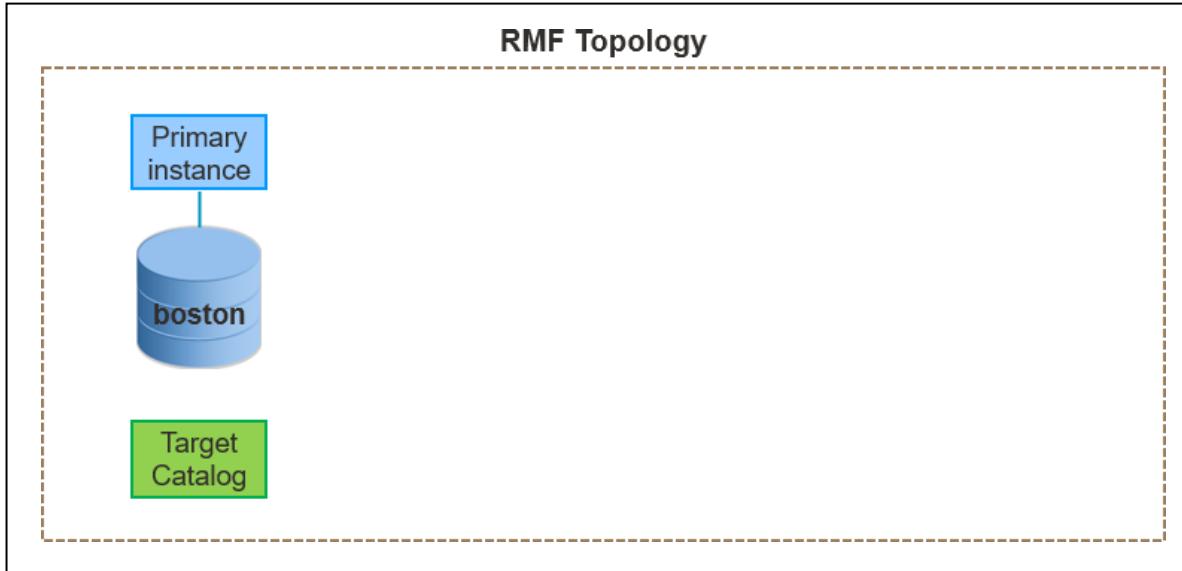
```

[oracle@host01 ~]$ cat /home/oracle/setup/AWR_s3.txt
echo "1.2 Create source node name at $stns"
sqlplus /nolog <<EOF
spool crumf_snnode.log
connect sys/oracle_4U@$ptns as sysdba
drop database link $pri2stby_link;
create database link $pri2stby_link connect to sys\$umf
identified by $sysumf_passwd using '$stns';
connect sys/oracle_4U@$stns as sysdba
alter system set "_umf_remote_enabled"=TRUE scope=BOTH;
exec dbms_umf.unconfigure_node;
exec dbms_umf.configure_node('$standby_name');
select dbms_umf.get_node_name_local as node_name from dual;
EOF
grep ORA- crumf_snnode.log | grep -v ORA-02024
if [ $? -eq 0 ]
then
echo "Error: Failed to create source node at $stns."
echo "Please check crumf_snnode.log for more detail"

```

```
exit 2
fi
[oracle@host01 ~]$
```

7. Review the `AWR_s4.txt` file. This step is to create the RMF topology.



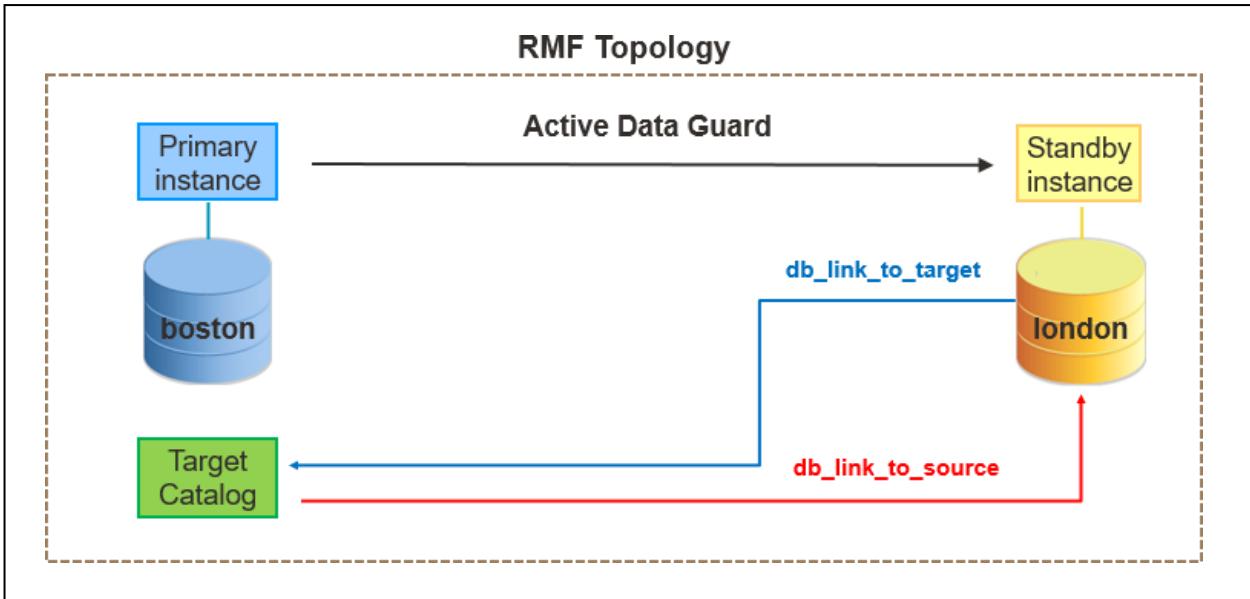
```
[oracle@host01 ~]$ cat /home/oracle/setup/AWR_s4.txt
echo "2. Create topology"
sqlplus /nolog <<EOF
spool crumf_topology.log
connect sys/oracle_4U@$ptns as sysdba
exec dbms_umf.drop_topology('$topology_alias');
exec dbms_umf.create_topology('$topology_alias');
-- Query the topology XML and X$#
select * from umf\$_topology_xml;
select * from x\$keumtoptb;
alter system archive log current;
EOF
grep ORA- crumf_topology.log | grep -v ORA-20507 | grep -v ORA-
06512| grep -v ORA-15767
if [ $? -eq 0 ]
then
echo "Error: Failed to create topology at $ptns."
echo "Please check crumf_topology.log for more detail"
exit 2
```

```

fi
sleep 5[oracle@host01 ~]

```

8. Review the AWR\_s5.txt file. This step is to register a node (london in this practice) with the RMF topology.

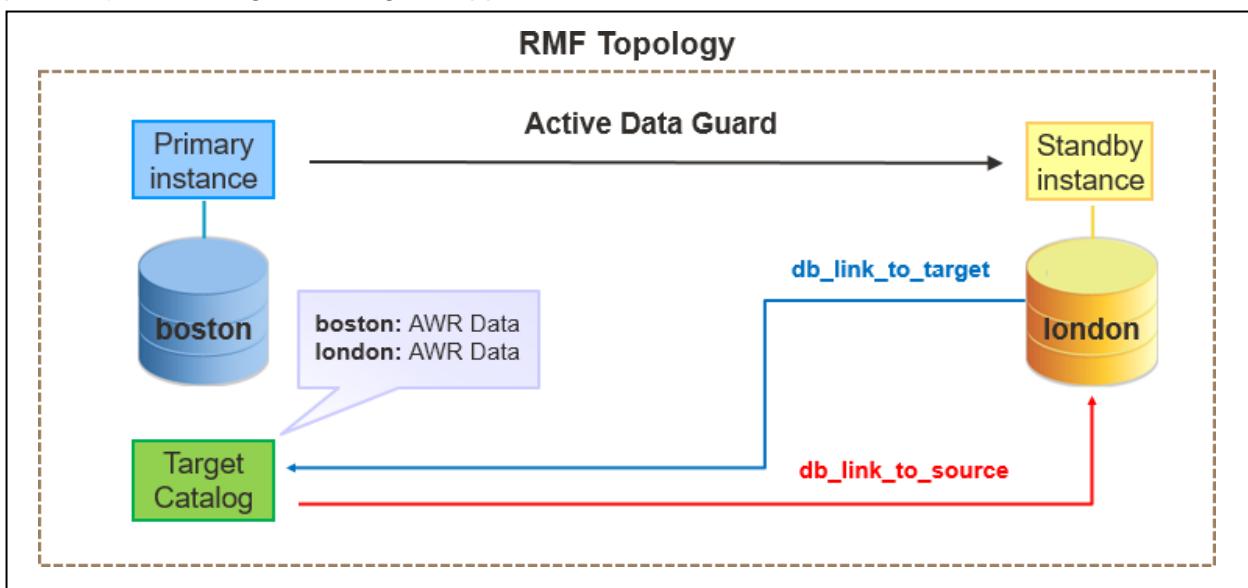


```

[oracle@host01 ~]$ cat /home/oracle/setup/AWR_s5.txt
echo "3. Register remote node at $stns"
sqlplus /nolog <<EOF
spool crumf_regregremote.log
connect sys/oracle_4U@$ptns as sysdba
set echo on;
exec dbms_umf.register_node('$topology_alias', '$standby_name',
'$pri2stby_link', '$stby2pri_link');
EOF
grep ORA- crumf_regregremote.log
if [ $? -eq 0 ]
then
  echo "Error: Failed to register remote node $stns at $ptns."
  echo "Please check crumf_regregremote.log for more detail"
  exit 2
fi
sleep 10[oracle@host01 ~]$

```

9. Review the `AWR_s6.txt` file. This step is to register a remote database (`london` in this practice) in the Target Catalog to support the remote AWR Data.



```
[oracle@host01 ~]$ cat /home/oracle/setup/AWR_s6.txt
echo "4. Register remote database at $stns"
sqlplus /nolog <<EOF
spool crumf_regremotedb.log
connect sys/oracle_4U@$ptns as sysdba
exec
dbms_workload_repository.register_remote_database('$standby_name')
);
EOF
grep ORA- crumf_regremotedb.log
if [ $? -eq 0 ]
then
echo "Error: Failed to register remote database $stns at $ptns."
echo "Please check crumf_regremotedb.log for more detail"
exit 2
fi
[oracle@host01 ~]$
```

10. Review the `AWR_s7` file. This step is to verify the configuration.

```
[oracle@host01 ~]$ cat /home/oracle/setup/AWR_s7.txt
echo "5. Verify setup"
sqlplus /nolog <<EOF
spool crumf_verify.log
```

```

connect sys/oracle_4U@$ptns as sysdba
select TOPOLOGY_NAME, NODE_NAME, NODE_ID, NODE_TYPE from
umf$\$registration;
select * from DBA_UMF_REGISTRATION;
-- Query x$'s.
select count(*) from x\$keumtoptb;
select count(*) from x\$keumregtb;
select count(*) from x\$keumlinktb;
select sysdate from dual@$pri2stby_link;
select dbid from v\$DATABASE@$stby2pri_link;
select * from sys.umf$\$topology@$stby2pri_link;
select sysdate from dual@$stby2pri_link;
select dbid from v\$DATABASE@$stby2pri_link;
select * from sys.umf$\$topology@$stby2pri_link;
-- Execute UMF/PLSQL API to query.

declare
topology_name VARCHAR2(128);
my_node_id NUMBER;
link_name VARCHAR2(128);
tid NUMBER;
begin
-- Query local node registration info. This will also sync.
dbms_umf.query_node_info(NULL, '$standby_name', my_node_id);
dbms_output.put_line('my_node_id='||my_node_id);
-- Get the target id.
select target_id into tid from umf$\$topology_xml where
topology_name =
'$topology_alias';
dbms_output.put_line('tid='||tid);
-- Query the link info.
dbms_umf.query_link_info('$topology_alias',my_node_id, tid ,
link_name);
dbms_output.put_line('link_name='||link_name);
end;
/
select target_id from umf$\$topology_xml where topology_name =
'$topology_alias';

```

```
EOF
```

```
[oracle@host01 ~]$
```

11. Now, you run the `AWR_ADG.sh` script to perform the steps you reviewed in this practice. Wait until the script execution is complete.

```
[oracle@host01 ~]$ /home/oracle/setup/AWR_ADG.sh
```

```
1. Create node
```

```
1.1 Create target node name at boston
```

```
SQL*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 15:48:41  
2020
```

```
Version 19.3.0.0.0
```

```
(c) 1982, 2019, Oracle. All rights reserved.SQL> SQL>
```

```
Connected.
```

```
SQL>
```

```
System altered.
```

```
SQL>
```

```
User altered.
```

```
SQL>
```

```
Database link dropped.
```

```
SQL>
```

```
Database link created.
```

```
SQL>
```

```
PL/SQL procedure successfully completed.
```

```
SQL>
```

```
PL/SQL procedure successfully completed.
```

```
SQL>
```

```
NODE_NAME
```

```
boston

SQL>
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.3.0.0.0

1.2 Create source node name at london

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 15:48:42
2020
Version 19.3.0.0.0

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

Connected.
SQL>
Database link dropped.

SQL>
Database link created.

SQL> Connected.
SQL>
System altered.

SQL>
PL/SQL procedure successfully completed.

SQL>
PL/SQL procedure successfully completed.

SQL>
NODE_NAME
-----
-----
london
```

```
SQL>
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.3.0.0.0
2. Create topology

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 15:48:44
2020
Version 19.3.0.0.0

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

Connected.
SQL>
PL/SQL procedure successfully completed.

SQL>
PL/SQL procedure successfully completed.

SQL> SQL>
TOPOLOGY_NAME
-----
-----
TARGET_ID TOPOLOGY_VERSION TOPOLOGY_STATE
-----
-----
Topology_Demo
1379144553          1          0

SQL>
ADDR           INDX      INST_ID      CON_ID
-----
-----
TNAME_KEUMTOPTB
-----
-----
TID_KEUMTOPTB VER_KEUMTOPTB ST_KEUMTOPTB
-----
```

00007FFA920A0E70	0	1	0
------------------	---	---	---

Topology\_Demo

1379144553	1	0
------------	---	---

SQL>

System altered.

SQL>

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

Version 19.3.0.0.0

**3. Register remote node at london**

SQL\*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 15:48:52  
2020

Version 19.3.0.0.0

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

Connected.

SQL> SQL>

PL/SQL procedure successfully completed.

SQL>

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

Version 19.3.0.0.0

**4. Register remote database at london**

SQL\*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 15:49:03  
2020

Version 19.3.0.0.0

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

Connected.

SQL>

```
PL/SQL procedure successfully completed.

SQL>
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.3.0.0.0

5. Verify setup

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 15:49:09
2020
Version 19.3.0.0.0

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

Connected.

SQL>
TOPOLOGY_NAME
-----
-----
NODE_NAME
-----
-----
NODE_ID  NODE_TYPE
-----
Topology_Demo
boston
1379144553      0

Topology_Demo
london
1659670991      0

TOPOLOGY_NAME
-----
-----
NODE_NAME
-----
-----
NODE_ID  NODE_TYPE
```

```
SQL>
TOPOLOGY_NAME
-----
NODE_NAME
-----
NODE_ID  NODE_TYPE AS_SO AS_CA STATE
-----
Topology_Demo
boston
1379144553      0 FALSE FALSE OK

Topology_Demo
london
1659670991      0 TRUE   FALSE OK

TOPOLOGY_NAME
-----
NODE_NAME
-----
NODE_ID  NODE_TYPE AS_SO AS_CA STATE
-----
SQL> SQL>
COUNT (*)
-----
1

SQL>
COUNT (*)
-----
```

2

```
SQL>  
  COUNT (*)
```

```
-----  
 2
```

```
SQL>  
SYSDATE
```

```
-----  
05-JUN-20
```

```
SQL>  
  DBID
```

```
-----  
2732274290
```

```
SQL>  
TOPOLOGY_NAME
```

```
-----  
-----  
TARGET_ID TOPOLOGY_VERSION TOPOLOGY_STATE
```

```
-----  
Topology_Demo  
1379144553          4          0
```

```
SQL>  
SYSDATE
```

```
-----  
05-JUN-20
```

```
SQL>  
  DBID
```

```
-----  
2732274290
```

```
SQL>
```

```

TOPOLOGY_NAME
-----
-----  

TARGET_ID TOPOLOGY_VERSION TOPOLOGY_STATE
-----  

Topology_Demo
1379144553          4          0  

SQL> SQL> 2   3   4   5   6   7   8   9   10  11  12
13   14   15   16   17   18
PL/SQL procedure successfully completed.  

SQL>
TARGET_ID
-----
1379144553  

SQL>
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.3.0.0.0
[oracle@host01 ~]$
```

12. In the same terminal session connected to host01, log in as the SYS user and create a remote snapshot.

```

[oracle@host01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Jun 10 18:01:58
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> exec  
dbms_workload_repository.create_remote_snapshot('london')
```

```
PL/SQL procedure successfully completed.
```

13. We need to have at least two remote snapshots to generate a report. Create another remote snapshot.

```
SQL> exec  
dbms_workload_repository.create_remote_snapshot('london')
```

```
PL/SQL procedure successfully completed.
```

14. Generate AWR report for the london standby database.

**Note:** If the awrrpt.sql script doesn't display the option for the instance section, then exit SQL\*Plus and log in again.

```
SQL> @?/rdbms/admin/awrrpti.sql
```

```
Specify the Report Type  
~~~~~
```

```
AWR reports can be generated in the following formats. Please  
enter the
```

```
name of the format at the prompt. Default value is 'html'.
```

```
'html'      HTML format (default)  
'text'      Text format  
'active-html'  Includes Performance Hub active report
```

```
Enter value for report_type: html
```

```
Type Specified: html
```

```
Instances in this Workload Repository schema  
~~~~~
```

DB Id	Inst Num	DB Name	Instance	Host
1659670991	1	BOSTON	london	host03.example.com
* 2732274290	1	BOSTON	boston	host01.example.com

Enter value for dbid: **1659670991**

Using 1659670991 for database Id

Enter value for inst\_num: **1**

Using 1 for instance number

Specify the number of days of snapshots to choose from

~~~~~

Entering the number of days (n) will result in the most recent (n) days of snapshots being listed. Pressing <return> without specifying a number lists all completed snapshots.

Enter value for num\_days: **1**

Listing the last day's Completed Snapshots

| Instance | DB Name | Snap Id | Snap Started      | Snap Level |
|----------|---------|---------|-------------------|------------|
| london   | BOSTON  | 1       | 05 Jun 2020 16:29 | 1          |
|          |         | 2       | 05 Jun 2020 16:31 |            |
| 1        |         |         |                   |            |

Specify the Begin and End Snapshot Ids

~~~~~

Enter value for begin\_snap: **1**

Begin Snapshot Id specified: 1

Enter value for end\_snap: **2**

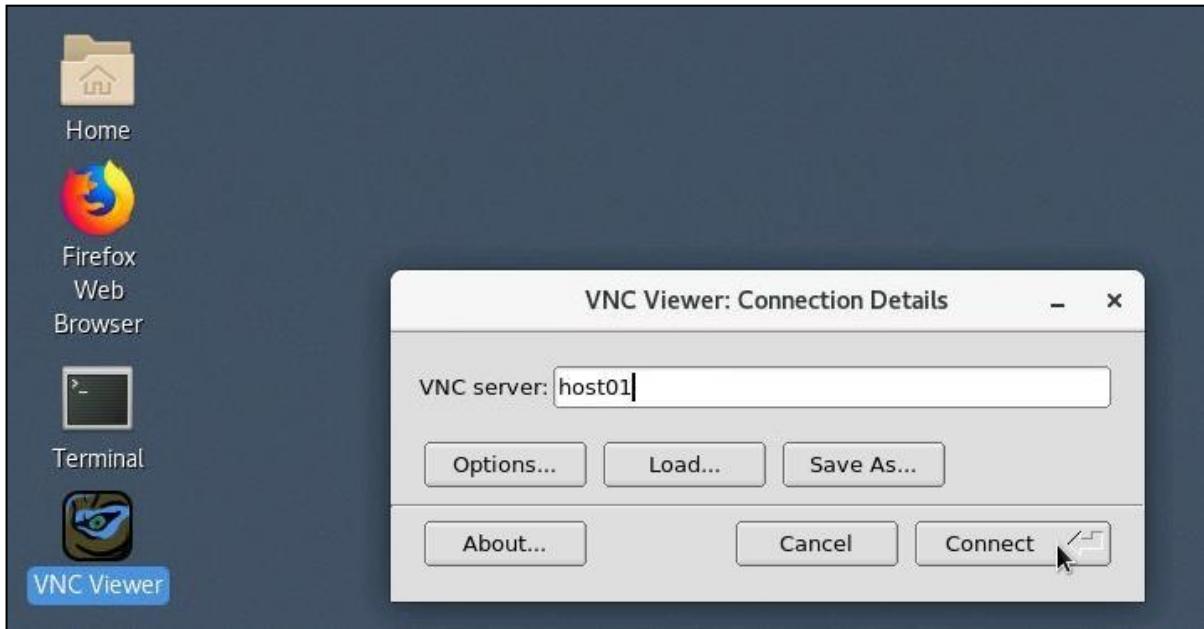
End Snapshot Id specified: 2

```
Specify the Report Name
~~~~~
The default report file name is awrrpt_1_1_2.html. To use this
name,
press <return> to continue, otherwise enter an alternative.

Enter value for report_name: awrrpt_1_1_2.html

Using the report name awrrpt_1_1_2.html
...
```

15. Click the VNC Viewer icon in the gateway VM and enter host01 in the VNC server field. Click **Connect**.



16. Select oracle and enter <password> in the Password field. Click Sign In.



17. Open a terminal window in the VNC viewer session and type `firefox awrrpt_1_1_2.html` to view the AWR report.

A screenshot of a Firefox browser window. The address bar shows 'file:///home/oracle/awrrpt\_1\_1\_2.html'. The page content is an AWR report titled 'AWR Report for DB: BOSTON, Inst: london, Snaps: 1-2 - Mozilla Firefox (on host01.example.com)'. It includes a table with columns: DB Name, DB Id, Unique Name, Role, Edition, Release, RAC, CDB. The 'Role' column for the first row ('BOSTON') is highlighted with a red box and contains 'PHYSICAL STANDBY'. Another table below shows Instance, Inst Num, and Startup Time for the 'london' instance. A third table at the bottom lists Container DB Id, Container Name, and Open Time for three entries: 1717338523 (DEV1, 05-Jun-20 13:11), 1956346905 (PDB\$SEED, 05-Jun-20 13:10), and 2732274290 (CDB\$ROOT, 05-Jun-20 13:10).

18. Close the AWR report and Tiger VNC viewer session connected to host01.

19. Exit SQL\*Plus on host01 and host03 leaving the terminal windows open for future practices.

Congratulations! You successfully generated the AWR report for the `london` standby database.

## Practice 13-3: Using ADDM for an Active Data Guard Instance

---

### Overview

In this practice, you will generate the ADDM report to support analysis of the workloads in Active Data Guard database.

### Tasks

1. Use a terminal window logged in as `oracle` to `host01` with the environment variables set for `boston` appropriately. Launch SQL\*Plus on your primary database.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [boston] ? boston
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@host01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Jun 5 22:55:53
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>
```

2. Display the information on members in the topology created in the previous practice.

```
SQL> @/home/oracle/setup/view_topology.sql
...
SQL> select * from dba_umf_registration;

TOPOLOGY_NAME      NODE_NAME    NODE_ID      NODE_TYPE    AS_SO AS_CA STATE
-----  -----
Topology_Demo      boston       1379144553          0 FALSE FALSE OK
Topology Demo      london        1659670991          0 TRUE  FALSE OK
```

```

SQL> select * from dba_umf_service;

TOPOLOGY_NAME      NODE_ID SERVICE
-----
Topology_Demo     1659670991 AWR

SQL>

```

3. Display the snapshots of the london standby database.

```

SQL> SELECT dbid, snap_id FROM dba_hist_snapshot WHERE dbid =
'1659670991' ORDER BY 2;

DBID      SNAP_ID
-----
1659670991    1
1659670991    2

8 rows selected.

SQL>

```

4. Use ADDM to analyze the workloads in the last two remote snapshots period.

```

SQL> var tname VARCHAR2(60);
SQL> BEGIN
  :tname := 'my_database_analysis_mode_task';
  DBMS_ADDM.ANALYZE_DB(:tname, 1, 2, db_id => '1659670991');
end;
/
2      3      4      5

PL/SQL procedure successfully completed.

SQL>

```

5. Generate the ADDM report. The report is generated for the physical standby database. Exit SQL\*Plus.

**Note:** The ADDM report below is a sample report. Your output may not include any findings.

```
SQL> @/home/oracle/setup/get_adm.sql
...
SQL> SELECT DBMS_ADDM.GET_REPORT(:tname) FROM DUAL;

DBMS_ADDM.GET_REPORT(:TNAME)
-----
-----
      ADDM Report for Task 'my_database_analysis_mode_task'
-----
-----
Analysis Period
-----
AWR snapshot range from 7 to 8.
Time period starts at 05-JUN-20 10.01.01 PM
Time period ends at 05-JUN-20 11.00.25 PM

Analysis Target
-----
Database 'BOSTON' with DB ID 1659670991.
Database version 19.0.0.0.0.
Analysis was requested for all instances, but ADDM analyzed
instance london,
numbered 1 and hosted at host03.example.com.
See the "Additional Information" section for more information on
the requested
instances.
ADDM detected that the database was open in READ-ONLY mode.
ADDM detected that the database type is MULTitenant DB.

Activity During the Analysis Period
-----
Total database time was 311 seconds.
The average number of active sessions was .09.
ADDM analyzed 1 of the requested 1 instances.
```

## Summary of Findings

Description	Active Sessions		Recommendations
	Percent of Activity		
<hr/>			
1 Hard Parse	.03	28.67	0
2 Shared Pool Latches	.01	16.7	0
3 Soft Parse	.01	15.83	2
4 I/O Throughput	0	4.42	1
5 PL/SQL Compilation	0	2.96	1
6 Unusual "Network" Wait Event	0	2.77	1
7 Session Connect and Disconnect	0	2.45	1

~~~~~  
~~~~~  
~~~~~  
~~~~~

## Findings and Recommendations

### Finding 1: Hard Parse

Impact is .03 active sessions, 28.67% of total activity.

-----  
Hard parsing of SQL statements was consuming significant database time.

Hard parses due to cursor environment mismatch were not consuming significant database time.

Hard parsing SQL statements that encountered parse errors was not consuming significant database time.

Hard parses due to literal usage and cursor invalidation were not consuming significant database time.

No recommendations are available.

Finding 2: Shared Pool Latches

Impact is .01 active sessions, 16.7% of total activity.

Contention for latches related to the shared pool was consuming significant database time.

Waits for "library cache lock" amounted to 1% of database time.

Waits for "cursor: pin S wait on X" amounted to 13% of database time.

No recommendations are available.

Symptoms That Led to the Finding:

Wait class "Concurrency" was consuming significant database time.

Impact is .01 active sessions, 16.76% of total activity.

Finding 3: Soft Parse

Impact is .01 active sessions, 15.83% of total activity.

Soft parsing of SQL statements was consuming significant database time.

Recommendation 1: Application Analysis

Estimated benefit is .01 active sessions, 15.83% of total activity.

Action

Investigate application logic to keep open the frequently used cursors.

Note that cursors are closed by both cursor close calls and session disconnects.

Recommendation 2: Database Configuration

Estimated benefit is .01 active sessions, 15.83% of total activity.

Action

Consider increasing the session cursor cache size by increasing the value of parameter "session\_cached\_cursors".

Rationale

The value of parameter "session\_cached\_cursors" was "50" during the analysis period.

Finding 4: I/O Throughput

Impact is 0 active sessions, 4.42% of total activity.

The throughput of the I/O subsystem was significantly lower than expected.

Recommendation 1: Host Configuration

Estimated benefit is 0 active sessions, 4.42% of total activity.

--

Action

Consider increasing the throughput of the I/O subsystem. Oracle's recommended solution is to stripe all data files using the SAME methodology. You might also need to increase the number of disks for better performance.

Rationale

During the analysis period, the average data files' I/O throughput was 0

K per second for reads and 0 K per second for writes. The average response time for single block reads was 0 milliseconds.

Symptoms That Led to the Finding:

Wait class "User I/O" was consuming significant database time.

Impact is 0 active sessions, 4.42% of total activity.

Finding 5: PL/SQL Compilation

Impact is 0 active sessions, 2.96% of total activity.

PL/SQL compilation consumed significant database time.

Recommendation 1: Application Analysis

Estimated benefit is 0 active sessions, 2.96% of total activity.

---

--

Action

Investigate the appropriateness of PL/SQL compilation. PL/SQL compilation can be caused by DDL on dependent objects.

Finding 6: Unusual "Network" Wait Event

Impact is 0 active sessions, 2.77% of total activity.

---

Wait event "SQL\*Net message from dblink" in wait class "Network" was consuming significant database time.

Recommendation 1: Application Analysis

Estimated benefit is 0 active sessions, 2.77% of total activity.

---

--

Action

Investigate the cause for high "SQL\*Net message from dblink" waits.

Refer to Oracle's "Database Reference" for the description of this wait event.

Symptoms That Led to the Finding:

---

Wait class "Network" was consuming significant database time.

Impact is 0 active sessions, 2.94% of total activity.

Finding 7: Session Connect and Disconnect

Impact is 0 active sessions, 2.45% of total activity.

---

Session connect and disconnect calls were consuming significant database time.

Recommendation 1: Application Analysis

Estimated benefit is 0 active sessions, 2.45% of total activity.

---

--

Action

Investigate application logic for possible reduction of connect and disconnect calls. For example, you might use a connection pool scheme in the middle tier.

---

---

---

---

Additional Information

---

Miscellaneous Information

---

Wait class "Application" was not consuming significant database time.

Wait class "Commit" was not consuming significant database time.  
Wait class "Configuration" was not consuming significant database time.

CPU was not a bottleneck for the instance.

SQL> **exit**

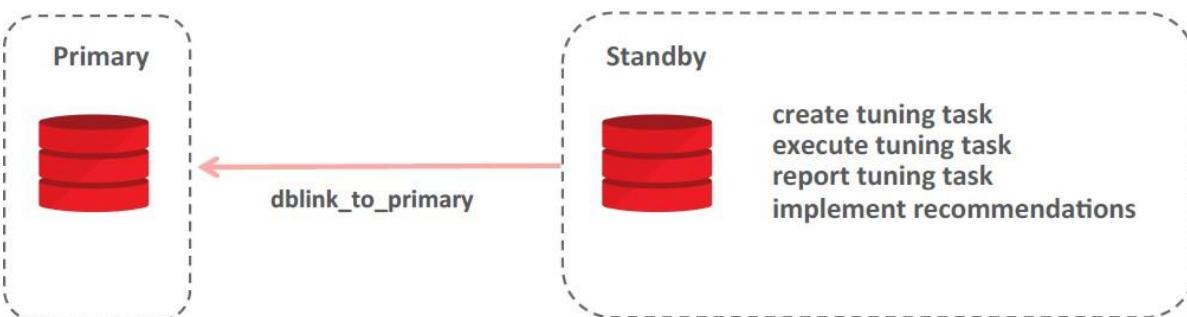
Disconnected from Oracle Database 19c Enterprise Edition Release  
19.0.0.0.0 - Production  
Version 19.3.0.0.0  
[oracle@host01 ~]\$

## 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.PART`, `OE.SUPPLIER`, `OE.DATE_DIM`, `OE.CUSTOMER`, and `OE.LINEORDER` tables.

```
[oracle@host01 ~]$ /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@host01 ~]$
```

2. Use a terminal window logged in as oracle to host01 with the environment variables set for boston appropriately. Log in to the DEV1 PDB as the SYS user and run the setup13-4.sql script.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@host01 ~]$ sqlplus
sys/<password>@host01:1521/dev1.example.com 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','lineorder');

PL/SQL procedure successfully completed.

SQL> exec dbms_stats.delete_table_stats('OE','date_dim');

PL/SQL procedure successfully completed.

SQL>
```

3. 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 'boston';
```

```
Database link created.
```

```
SQL>
```

4. Use a terminal window logged in as `oracle` to `host03` with the environment variables set for `london` appropriately. Launch SQL\*Plus and run the problem query in the `DEV1` PDB.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@host03 ~]$ 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
SQL> SELECT /* problem_query */
 2   SUM(lo_extendedprice * lo_discount) revenue
```

```

3  FROM oe.lineorder l, oe.date_dim d
4  WHERE l.lo_orderdate = d.d_datekey;

REVENUE
-----
3.2632E+13

SQL>

```

5. 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(lo_extendedprice * lo_discount)
revenue FROM oe.lineorder l, oe.date_dim d WHERE l.lo_orderdate = d.d_datekey

SQL>

```

6. Open a new terminal window logged in as oracle to host03 with the environment variables set for london appropriately.

```

[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been set to /u01/app/oracle
[oracle@host03 ~]$ sqlplus / as sysdba

```

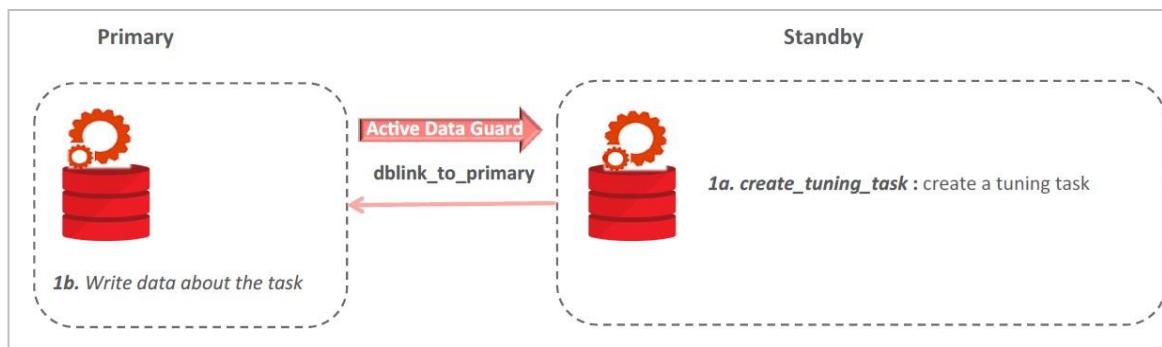
```
SQL*Plus: Release 19.0.0.0.0 - Production on Thu Jun 11 01:19:22  
2020  
Version 19.3.0.0.0  
  
(c) 1982, 2019, Oracle. All rights reserved.  
  
Connected to:  
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -  
Production  
Version 19.3.0.0.0  
  
SQL>
```

7. 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(lo_extendedprice * lo_discount)  
revenue FROM oe.l  
ineorder l, oe.date_dim d WHERE l.lo_orderdate = d.d_datekey  
  
SQL>
```

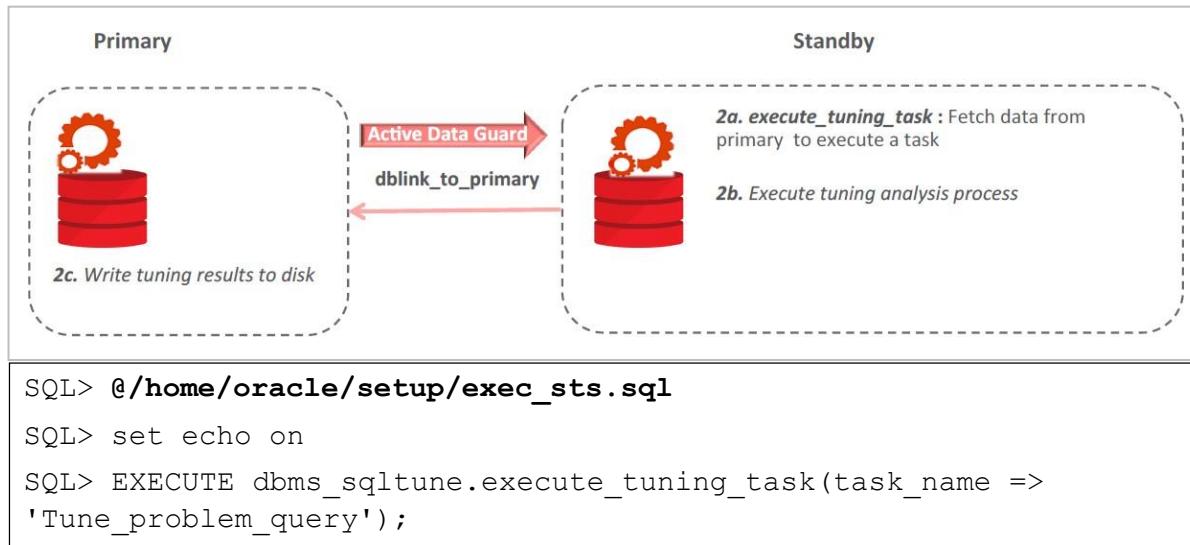
## 8. Create a SQL Tuning Task.

**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
  2      stmt_task VARCHAR2(64);
  3  BEGIN
  4      stmt_task:=dbms_sqltune.create_tuning_task(sql_id =>
'an7zryzf86prm', task_name => 'Tune_problem_query',
database_link_to => 'DBLINK_TO_PRIMARY.EXAMPLE.COM');
  5  END;
  6  /
PL/SQL procedure successfully completed.
```

## 9. Execute the SQL Tuning Task.

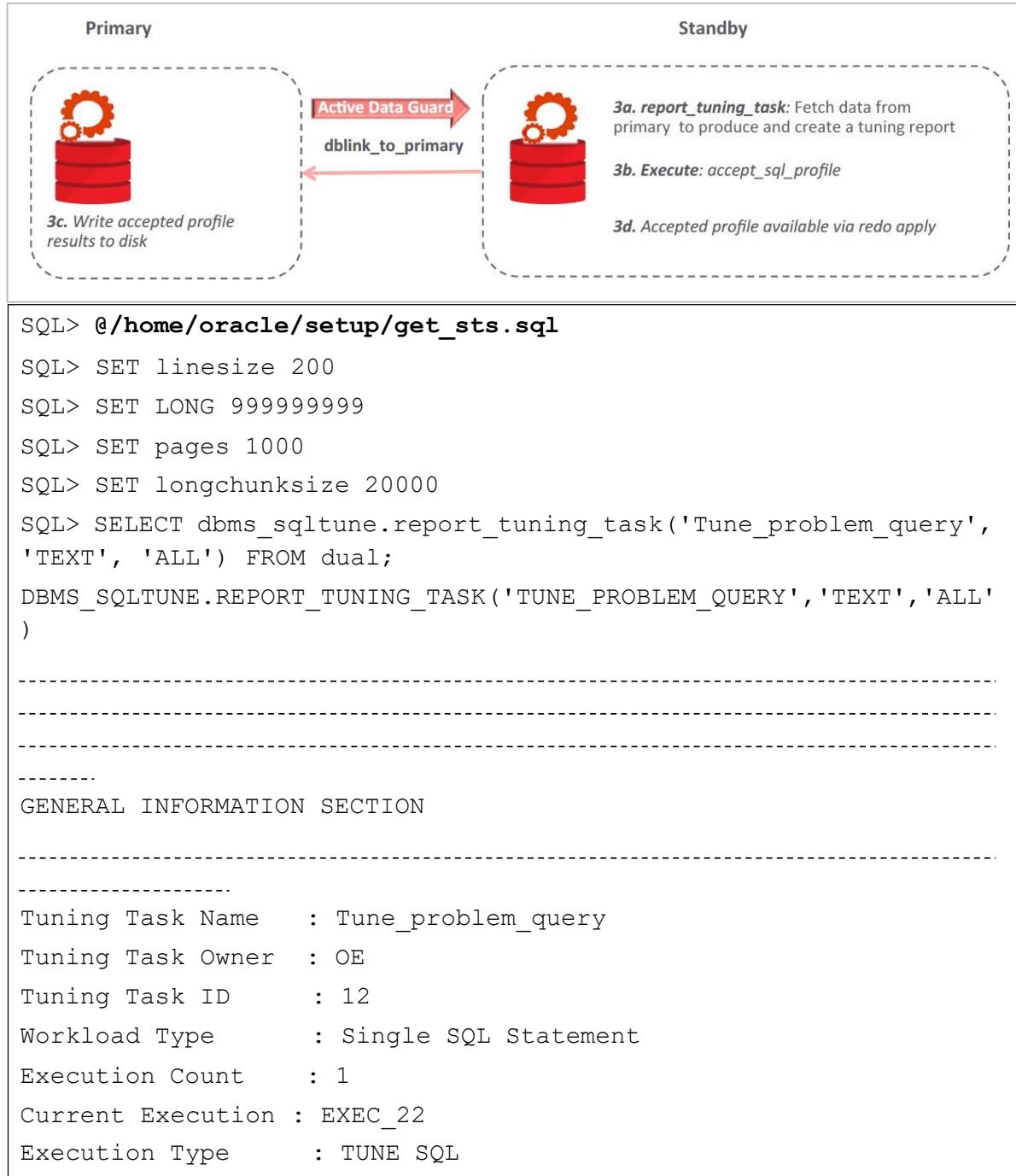


PL/SQL procedure successfully completed.

SQL>

## 10. Generate the SQL Tuning Task report.

**Note:** The result varies.



```
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.
```

## 2- Statistics Finding

---

Table "OE"."LINEORDER" was not analyzed.

### Recommendation

---

- Consider collecting optimizer statistics for this table.

```
execute dbms_stats.gather_table_stats(ownname => 'OE',
tabname =>
    'LINEORDER', 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					
(1)	SELECT STATEMENT		1	52	6448 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
3 - SEL$1 / D@SEL$1
4 - 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>
```

11. Return to the terminal session connected to host01. Switch to the DEV1 container.

```
SQL> show con_name

CON_NAME
-----
CDB$ROOT
SQL>
SQL>
SQL> alter session set container=dev1;

Session altered.

SQL>
```

12. 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 => 'DATE_DIM', 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 => 'LINEORDER', estimate_percent =>
  DBMS_STATS.AUTO_SAMPLE_SIZE, method_opt => 'FOR ALL COLUMNS SIZE
  AUTO');

PL/SQL procedure successfully completed.

SQL>
```

13. 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 host01 connected as oracle with the environment variables set to boston. Launch the DGMGRL utility and connect as the SYSdg user.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$ 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>@boston
Connected to "boston"
Connected as SYSDG.
DGMGRL>
```

2. Use the SHOW CONFIGURATION command to display the configuration status for the Data Guard configuration.

```
DGMGRL> show configuration

Configuration - DRSolution

Protection Mode: MaxPerformance
Members:
boston   - Primary database
bostonFS - Far sync instance
london   - Physical standby database
london2  - Logical standby database

Members Not Receiving Redo:
londonFS - Far sync instance
```

```
Fast-Start Failover: Disabled

Configuration Status:
SUCCESS      (status updated 21 second ago)

DGMGRL>
```

3. 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 boston

Database Role:      Primary database

Ready for Switchover: Yes

Flashback Database Status:
    boston: Off

Capacity Information:
    Database Instances          Threads
    boston      1                  1

Managed by Clusterware:
    boston: NO
    Validating static connect identifier for the primary
database boston...
    The static connect identifier allows for a connection to
database "boston".

Temporary Tablespace File Information:
    boston TEMP Files: 3

Data file Online Move in Progress:
    boston: No

Transport-Related Information:
    Transport On: Yes

Log Files Cleared:
    boston Standby Redo Log Files: Cleared

DGMGRL>
```

4. Validate that the physical standby database is ready for role reversal using the VERBOS option.

```
DGMGRL> validate database verbose london

Database Role:      Physical standby database
Primary Database:   boston

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

Flashback Database Status:
  boston: Off
  london: Off

Capacity Information:
  Database Instances          Threads
  boston        1                  1
  london        1                  1

Managed by Clusterware:
  boston: NO
  london: NO
  Validating static connect identifier for the primary
  database boston...
  The static connect identifier allows for a connection to
  database "boston".

Temporary Tablespace File Information:
  boston TEMP Files: 3
  london TEMP Files: 3

Data file Online Move in Progress:
  boston: No
  london: 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:

boston Standby Redo Log Files: Cleared

london Online Redo Log Files: Cleared

london Standby Redo Log Files: Available

Current Log File Groups Configuration:

Thread #	Online Redo Log Groups	Standby Redo Log Groups
----------	------------------------	-------------------------

Status

(boston)

(london)

1 3

2

Insufficient SRLs

Future Log File Groups Configuration:

Thread #	Online Redo Log Groups	Standby Redo Log Groups
----------	------------------------	-------------------------

Status

(london)

(boston)

1 3

0

Insufficient SRLs

Warning: standby redo logs not configured for thread 1 on boston

Current Configuration Log File Sizes:

Thread #	Smallest Online Redo	Smallest Standby Redo
----------	----------------------	-----------------------

Log File Size

Log File Size

(boston)

(london)

1 200 MBytes

200 MBytes

Apply-Related Property Settings:

Property	boston Value	london Value
DelayMins	0	0
ApplyParallel	AUTO	AUTO
ApplyInstances	0	0

Transport-Related Property Settings:

Property	boston Value	london 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>

5. Use the terminal connected to host03 as oracle with the environment variables set to london. Launch SQL\*Plus and connect as the SYSDG user.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been set to /u01/app/oracle
[oracle@host03 ~]$ 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>
```

6. Check the value of the STANDBY\_DB\_PRESERVE\_STATES parameter.

SQL> show parameter STANDBY_DB_PRESERVE_STATES		
NAME	TYPE	VALUE
standby_db_preserve_states	string	NONE
SQL>		

**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.

7. 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>
```

8. Return to the DGMGRL session on host01. Switch over to the london physical standby database.

```
DGMGRL> switchover to london
Performing switchover NOW, please wait...
Operation requires a connection to database "london"
Connecting ...
Connected to "london"
Connected as SYSDG.
New primary database "london" is opening...
Operation requires start up of instance "boston" on database
"boston"
Starting instance "boston"...
Connected to an idle instance.
ORACLE instance started.
Connected to "boston"
Database mounted.
Database opened.
Connected to "boston"
Switchover succeeded, new primary is "london"
DGMGRL>
```

9. 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:
```

```
london      - Primary database
londonFS   - Far sync instance
boston     - Physical standby database
london2    - Logical standby database

Members Not Receiving Redo:
bostonFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS      (status updated 21 second ago)

DGMGRL> exit
[oracle@host01 ~]$
```

**Note:** Remember that the indentation used in the output of the SHOW CONFIGURATION command indicates the hierarchy of how redo is being forwarded.

10. Return to the terminal session connected to host03. 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@host03 ~]$
```

**Note:** The physical standby session was lost during role transition. This is the default behavior.

11. In the same terminal window on host03, launch the DGMGRL utility and connect as the SYSDG user.

```
[oracle@host03 ~]$ 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>@london
Connected to "london"
Connected as SYSDG.
```

12. Perform a log switch on the new primary database london from within DGMGRL.

```
DGMGRL> SQL "alter system switch logfile";
Succeeded.
```

13. Verify that the new standby database boston has zero transport lag and zero apply lag. You may need to wait a minute for this to clear.

```
DGMGRL> show database boston

Database - boston

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

Database Status:
SUCCESS

DGMGRL>
```

14. 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

- With the navigation techniques learned in practice 10-3, navigate to the Databases pages.

The screenshot shows the Oracle Enterprise Manager Cloud Control 13c interface. The title bar reads "ORACLE® Enterprise Manager Cloud Control 13c". The main menu has tabs for Performance, Availability, Security, Schema, and Administration. A search bar at the top right includes a "Find Name" input field and a magnifying glass icon. Below the search bar are buttons for View, Add, Remove, and Configure. The main content area is titled "Databases" and displays a table with the following data:

Name	Type	Status	Target Version	Incidents	Average Compliance Score	Member Status Summary
boston.example.com	Database Instance : Container, Primary	Green	19.3.0.0.0	0 0 0	N/A	0 1 0 0
ORCL	Database Instance : Container	Green	19.3.0.0.0	0 0 0	N/A	0 1 0 0
rcatcdb	Database Instance : Container	Green	19.3.0.0.0	0 0 0	N/A	0 2 0 0

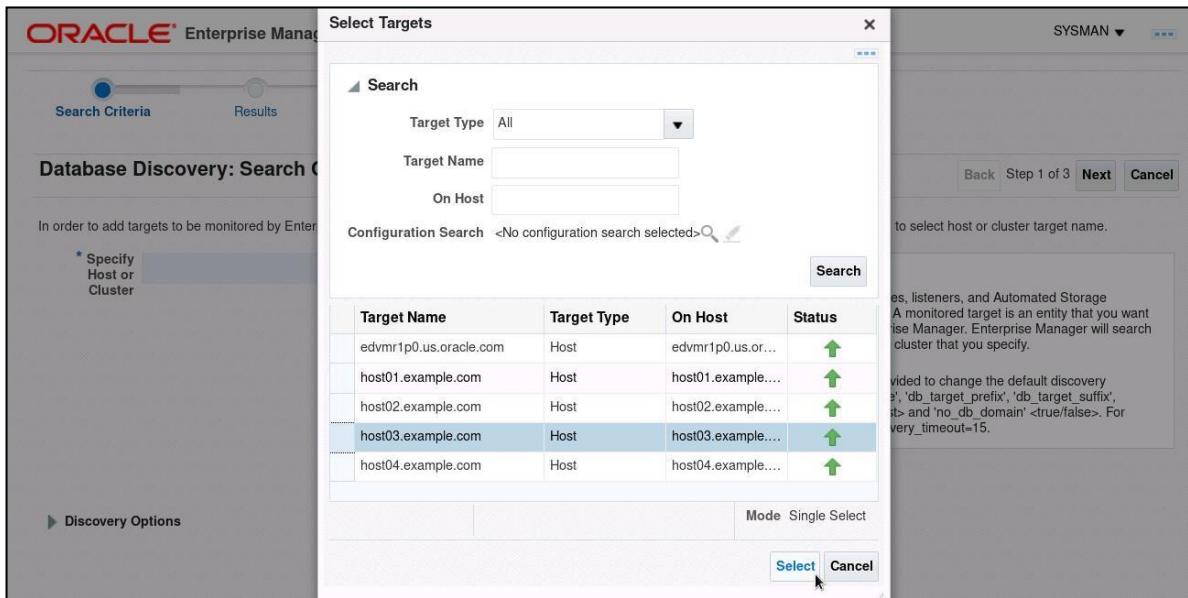
- Add the `london` database as an EM target in preparation of the Switchover practice through Enterprise Manager. Click **Add > Oracle Database**.

The screenshot shows the Oracle Enterprise Manager Cloud Control 13c interface, specifically the "Oracle Database" list page. The title bar and menu are identical to the previous screenshot. The main content area shows a table with the following data:

Name	Type	Status
boston.example.com	Database Instance : Container, Physica...	Green
ORCL	Database Instance : Container	Green
rcatcdb	Database Instance : Container	Green

The "Add" button is highlighted with a blue box and a cursor arrow pointing to it.

3. On the Database Discovery: Search Criteria page, enter the name of the host (host03) by clicking the magnifying glass icon and selecting the host name in the dialog box. Then click **Next** to proceed.



4. On the Database Discovery: Results page, select the `london` 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.

Databases					
The following databases have been discovered on this host. Provide monitoring credentials and save the targets to start monitoring the databases. You can specify common monitoring credentials for all the selected database targets using the 'Specify Common Monitoring Credentials' action. You can set Global Target Properties for all selected targets or add them to a Target Group while saving the targets for monitoring.					
View ▾		Specify Common Monitoring Credentials		Configure Test Connection	
		Monitoring Credentials		Target Group	
<input type="checkbox"/>	Target Name	Monitor Username	Monitor Password	Role	
<input type="checkbox"/>	▶ london2.example.com (Container Data)	dbsnmp		Normal	<input type="button" value=""/>
<input checked="" type="checkbox"/>	▶ london.example.com (Container Data)	sys	*****	SYSDBA	<input type="button" value=""/>

5. Select the listed listener on `host03`. Click **Next**.

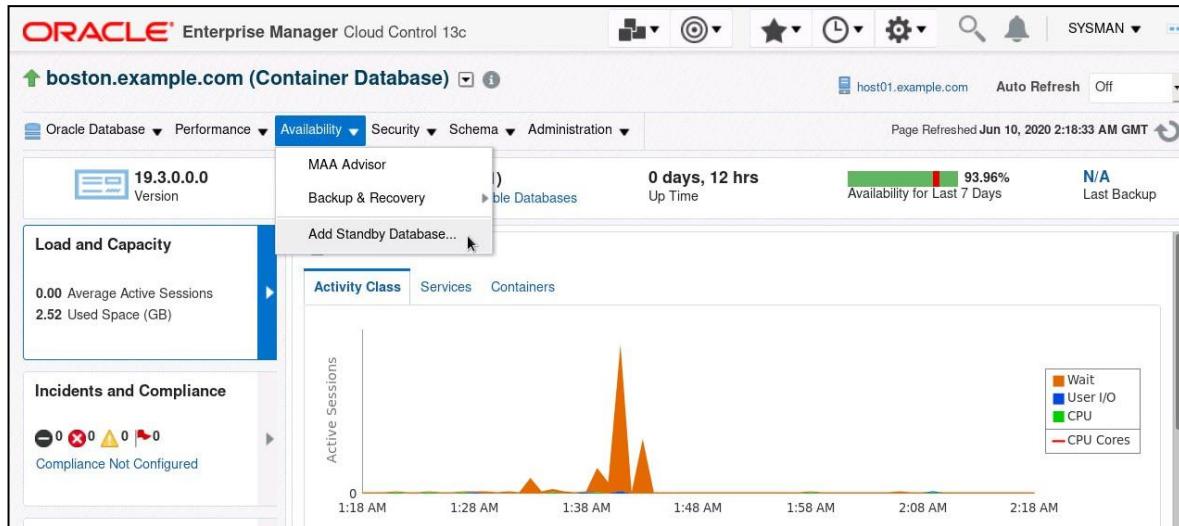
Listeners				
The following listeners have been discovered on this host.				
View ▾		Configure		
<input type="checkbox"/>	Target Name	Listener Name	Machine Name	Target Group
<input checked="" type="checkbox"/>	LISTENER_host03.example.com	LISTENER	host03.example.com	<input type="button" value=""/>

6. On the Database Discovery: Review page, click **Save**.
7. In the Confirmation dialog box, click **Close**.
8. On the Databases page, set the Auto Refresh option to 30 seconds and wait until the `london.example.com` target becomes normal. Click the link for the `boston.example.com` target.

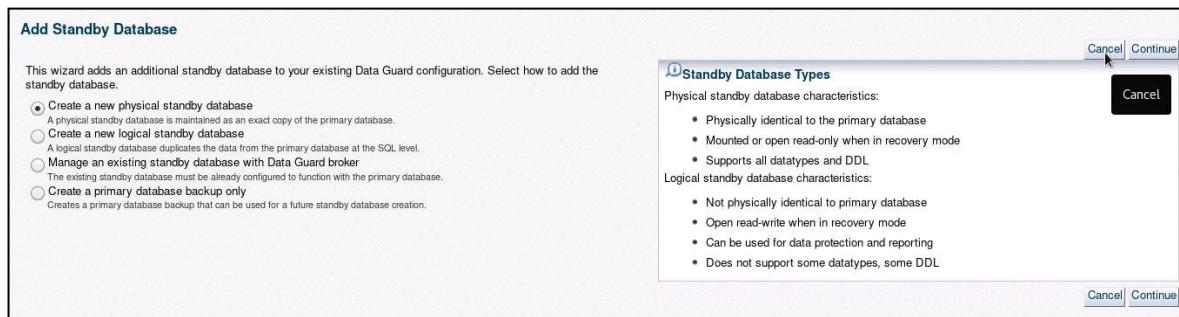
Name	Type	Status	Target Version	Incidents
boston.example.com	Database Instance : Container, Physica...		19.3.0.0.0	0 3 0
london.example.com	Database Instance : Container, Primary		19.3.0.0.0	0 0 0
ORCL	Database Instance : Container		19.3.0.0.0	0 0 0
createdb	Database Instance : Container		19.3.0.0.0	0 0 0

9. On the `boston.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\_BOSTON\_SYS2

10. On the Data Guard home page, make sure that the status of the current primary and physical standby is Normal.

The screenshot shows the Oracle Enterprise Manager Cloud Control 13c interface for a container database named 'boston.example.com'. The 'Data Guard' section displays the following information:

- Primary Database:** Name: london.example.com, Host: host03.example.com, Data Guard Status: Normal (highlighted with a red box).
- Standby Databases:** A table listing four databases:
 

Select	Name	Host	Data Guard Status	Role
<input checked="" type="radio"/>	boston.example.com	host01.example.com	Normal	Physical Standby
<input type="radio"/>	bostonFS	host02.example.com	Normal	Far Sync
<input type="radio"/>	london2.example.com	host03.example.com	Normal	Logical Standby
<input type="radio"/>	londonFS	host04.example.com	Normal	Far Sync

11. Use the terminal connected to host01 as oracle with the environment variables set to boston. Launch SQL\*Plus and connect as the SYS user.

```
[oracle@host01 ~] $ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~] $ 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>
```

12. Set the value of the STANDBY\_DB\_PRESERVE\_STATES parameter to SESSION and restart the boston 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  
-----  
 2 PDB$SEED           READ ONLY  NO  
 3 DEV1               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.

13. 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@host01 ~]$
```

14. Now, let's establish a new session for testing.

```
[oracle@host01 ~]$ sqlplus
oe/<password>@host01:1521/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>

```

15. With the navigation techniques learned, return to the EM Data Guard Home page. Switch over to the `boston` physical standby database like the following.

**Note:** If the EM page shows a warning message due to the restart of the `boston` database, refresh the browser.

- On the EM Data Guard Home page, select `boston.example.com` in the Standby Databases section. Click **Switchover**.

Standby Databases				
Select	Name	Host	Data Guard Status	Role
<input checked="" type="radio"/>	boston.example.com	host01.example.com	<span style="color: green;">✓ Normal</span>	Physical Standby
<input type="radio"/>	bostonFS	host02.example.com	<span style="color: green;">✓ Normal</span>	Far Sync
<input type="radio"/>	london2.example.com	host03.example.com	<span style="color: green;">✓ Normal</span>	Logical Standby
<input type="radio"/>	londonFS	host04.example.com	<span style="color: green;">✓ Normal</span>	Far Sync

- On the Host Login (`host01.example.com`) page, select **Preferred** in the Credential option. Click **Continue**.

Host Login: host01.example.com (Container Database) ●

Logged in as sys 🔒 | host01.example.com

Host Login: host01.example.com

Supply operating system credentials for a user who can access the Oracle Home for database boston.example.com on host host01.example.com.

Credential  Preferred  Named  New

Preferred Credential Name Database Host Credentials

Attribute	Value
UserName	oracle
Password	*****

More Details

Cancel Continue

- c. On the Host Login (`host03.example.com`) page, select **Preferred** in the Credential option. Click **Continue**.

- d. On the Confirmation page, click **Yes**.

- e. Monitor the progress of Switchover. **DON'T WAIT** for completion. Move on to the next step.

16. Return to the SQL\*Plus session connected on `host01`. Check the current status of the `OE` session periodically. Exit SQL\*Plus.

```
SQL> /
-----
      USERNAME          SID  SERIAL#
-----
          OE              42    40779
-----

SQL> /
-----
      USERNAME          SID  SERIAL#
-----
          OE              42    40779
-----
```

```

USERNAME          SID  SERIAL#
-----
OE                42    40779

SQL> /

USERNAME          SID  SERIAL#
-----
OE                42    40779

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition
Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
[oracle@host03 ~]$

```

**Note:** The OE session hangs for a while and resumes. With the new feature, the session is retained during role transition.

17. Return to the EM page. Once the switchover operation is complete, you will see the new primary database (`boston`) on the Data Guard home page.

The screenshot shows the Oracle Enterprise Manager (EM) interface for Data Guard. At the top, there is a yellow banner with an information icon and the text "Information: Switchover completed successfully." Below this, the title "Data Guard" is displayed, followed by the text "Page Refreshed June 10, 2020 9:03:50 PM UTC".

The main content area is titled "Overview" and displays the following status information:

- Data Guard Status: ✓ Normal
- Protection Mode: Maximum Performance
- Fast-Start Failover: Disabled

Below the overview, there is a section titled "Primary Database" which lists the following details for the database "boston.example.com" (which has a red box around its name):

- Name: boston.example.com
- Host: host01.example.com
- Data Guard Status: ✓ Normal
- Current Log: 86
- Properties: Edit

# **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 host01 connected as oracle with the environment variables set to boston. Launch SQL\*Plus and determine the current state of flashback database.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$ 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>
```

- 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>
```

- 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      string      /u01/app/oracle
                             /fast_recovery_area
db_recovery_file_dest_size big integer 15000M
```

- 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
SQL>
```

- 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.
```

- Enable flashback database for the whole database.

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

```
Database altered
```

7. Verify that flashback database has been enabled.

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

FLASHBACK_ON
-----
YES
```

8. Determine the current size (in bytes) of the flashback data.

```
SQL> select flashback_size from v$flashback_database_log;

FLASHBACK_SIZE
-----
419430400

SQL>
```

9. 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/BOSTON/flashback/o1_mf_hfqbw2
0q_.flb
209715200

/u01/app/oracle/fast_recovery_area/BOSTON/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@host01 ~]$
```

## 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 host03 connected as oracle with the environment variables set to london. Launch SQL\*Plus and determine the current state of the flashback database.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been set to /u01/app/oracle
[oracle@host03 ~]$ 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>
```

2. Verify that the physical 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      0
Next log sequence to archive   0
Current log sequence          0
SQL>

```

- 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      string      /u01/app/oracle
                             /fast_recovery_area
db_recovery_file_dest_size big integer 15000M

```

- 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

```

- 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.

```

- 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

```

- Stop the managed recovery mode for the physical standby database.

```

SQL> alter database recover managed standby database cancel;
Database altered.

```

8. Return to the SQL\*Plus session on host03 connected to the london physical standby database and enable flashback database a second time.

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

```
Database altered.
```

9. Verify that flashback database has been enabled.

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

```
FLASHBACK_ON
```

```
-----  
YES
```

10. Restart the managed recovery mode for the london 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@host03 ~]$
```

## 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 host03 connected as oracle with the environment variables set to london2. Launch SQL\*Plus and determine the current state of flashback database.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [london] ? london2
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@host03 ~]$ 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>
```

- 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>
```

- 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      string      /u01/app/oracle
                             /fast_recovery_area
db_recovery_file_dest_size big integer 15000M
```

- 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
```

- 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.
```

- Enable flashback database for the whole database. Note the error message that is returned.

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

7. 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@host03 ~]$
```

## 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 host01 connected as oracle with the environment variables set to boston. Launch SQL\*Plus to connect to DEV1 PDB as the SYS user and create a guaranteed restore point called boston\_grp.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [boston] ? boston
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@host01 ~]$ sqlplus
sys/<password>@host01:1521/dev1.example.com 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 boston_grp GUARANTEE FLASHBACK DATABASE;

Restore point created.

SQL> col name format a30
SQL> SELECT name, scn, replicated FROM v$restore_point;

NAME                      SCN  REP
-----  -----
BOSTON_GRP                3955268 NO
```

2. 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

```

3. 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>

```

4. 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

```

5. Use a terminal window on host03 connected as oracle with the environment variables set to london. Launch SQL\*Plus to connect to DEV1 PDB. If the DEV1 PDB is not open, open it first.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been set to /u01/app/oracle
[oracle@host03 ~]$ 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
-----
 2 PDB$SEED              MOUNTED
 3 DEV1                  MOUNTED

-- If the london 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
-----
 2 PDB$SEED              READ ONLY NO
 3 DEV1                  READ ONLY NO

SQL>
SQL> alter session set container = DEV1;

Session altered.

SQL>
```

6. Verify that the restore points were replicated to the london standby database.

```
SQL> col name format a30
SQL> SELECT name, scn, replicated FROM v$restore_point;

NAME                      SCN REP
-----
BOSTON_GRP_PRIMARY          3955268 YES
```

**Note:** The restore point created in the primary database was replicated to the physical standby database.

7. Query the data in the HR schema in the london 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 london physical standby database.

8. Return to the terminal session connected to host01. Shut down and mount the boston 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@host01 ~]$
```

9. Log in to the RMAN utility to run the `FLASHBACK DATABASE` command to flash back the database to the restore point called `boston_grp`. Exit RMAN.

```
[oracle@host01 ~]$ 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: BOSTON (DBID=2732274290, not open)  
  
RMAN> FLASHBACK DATABASE TO RESTORE POINT BOSTON_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@host01 ~]$
```

10. Using SQL\*Plus, log in as the `SYS` user to open the primary database and the `DEV1` PDB in read-only mode.

```
[oracle@host01 ~]$ 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.
```

11. 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.

12. 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
----- -----
        2 PDB$SEED          READ ONLY NO
        3 DEV1              READ WRITE NO
SQL>

```

13. Return to the terminal session connected to host03 to check if the automatic flashback feature was used in the london 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.

14. In the same SQL\*Plus session on host03, review the alert log file. Press **CTRL+C** to exit.

```

SQL> host tail -100
/u01/app/oracle/diag/rdbms/london/london/trace/alert_london.log|more
...
Errors in file
/u01/app/oracle/diag/rdbms/london/london/trace/london_mrp0_26165.t
rc:
ORA-19909: datafile 1 belongs to an orphan incarnation
ORA-01110: data file 1:
' /u01/app/oracle/oradata/LONDON/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 (london)

```

```

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/LONDON/autobackup/2020_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.

15. Connect as the SYS user and mount the london 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          9140816 bytes
Variable Size        767557632 bytes
Database Buffers     96468992 bytes
Redo Buffers         7634944 bytes
Database mounted.
SQL>

```

16. Review the alert log file again. Press Ctrl+C to exit.

```

SQL> host grep -i flashback
/u01/app/oracle/diag/rdbms/london/london/trace/alert_london.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 london physical standby database.

17. Open the standby database and DEV1 PDB.

```
SQL> alter database open;
Database altered.

SQL> alter pluggable database dev1 open;
Pluggable database altered.

SQL>
```

18. 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.

19. Return to the SQL\*Plus session on host01 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 boston_grp;  
Restore point dropped.  
  
SQL>
```

20. Exit SQL\*Plus on host01 and host03 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 london2 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 london2 logical standby database manually.

### Tasks

1. Use the terminal window connected to host03 as oracle with the environment variables set to london2. Connect to the DEV1 PDB.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london2
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@host03 ~]$ 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>
```

2. 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>
```

3. Open a new terminal window connected to host03. Review the alert log file for the london2 standby database.

```
[oracle@host03 ~]$ tail -f  
/u01/app/oracle/diag/rdbms/london2/london2/trace/alert_london2.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/london2/london2/trace/london2_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.

4. 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 host01 as oracle.

```
[oracle@host01 ~]$ . oraenv  
ORACLE_SID = [boston] ? boston  
The Oracle base remains unchanged with value /u01/app/oracle  
[oracle@host01 ~]$ 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>
```

5. 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>
```

6. Return the SQL\*Plus session connected to the london2 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>
```

7. 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 9140816 bytes  
Variable Size 767557632 bytes  
Database Buffers 96468992 bytes  
Redo Buffers 7634944 bytes  
Database mounted.  
SQL> FLASHBACK DATABASE TO SCN <SCN in step 6>;  
  
Flashback complete.
```

8. Open the london2 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 flushed back.

9. Open the london2 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.
```

10. 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@host03 ~]$
```

11. Launch the DGMGRL utility and connect as the SYSDG user.

```
[oracle@host03 ~]$ 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>@boston
```

```
Connected to "boston"
```

```
Connected as SYSDG.
```

12. Use the SHOW CONFIGURATION command to display the configuration status for the Data Guard configuration.

```
DGMGRL> show configuration
```

```
Configuration - DRSSolution
```

```
Protection Mode: MaxPerformance
```

```
Members:
```

```
boston - Primary database
```

```
bostonFS - Far sync instance
```

```
london - Physical standby database
```

```
london2 - Logical standby database
```

```
Members Not Receiving Redo:  
londonFS - Far sync instance  
  
Fast-Start Failover: Disabled  
  
Configuration Status:  
SUCCESS (status updated 56 seconds ago)  
  
DGMGRL>
```

13. 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 host01 connected as oracle with the environment variables set to boston. Launch SQL\*Plus and perform a log switch on the primary database. Exit SQL\*Plus.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$
[oracle@host01 ~]$ 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@host01 ~]$
```

2. Use a terminal window on host02 connected as oracle with the environment variables set to bostonFS. Launch the DGMGRL utility and connect as the SYSDG user.

```
[oracle@host02 ~]$ . oraenv
ORACLE_SID = [oracle] ? bostonFS
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>@bostonFS
Connected to "bostonFS"
Connected as SYSDG.
DGMGRL>
```

3. 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 london

Database - london

Enterprise Manager Name: london.example.com
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: 7.00 KByte/s
Real Time Query: ON
Instance(s):
    london

Database Status:
SUCCESS

DGMGRL> show database london2

Database - london2

Enterprise Manager Name: london2.example.com
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):
    london2

Database Status:
SUCCESS
```

4. Display the current configuration and note the current state of fast-start failover.

```
DGMGRL> show configuration

Configuration - DRSSolution

Protection Mode: MaxPerformance
Members:
boston    - Primary database
bostonFS - Far sync instance
london    - Physical standby database
london2   - Logical standby database

Members Not Receiving Redo:
londonFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS      (status updated 46 seconds ago)

DGMGRL>
```

5. Set up the `FastStartFailoverTarget` configuration property on the `boston` primary database to indicate the desired `london` target standby database.

```
DGMGRL> edit database boston set property
FastStartFailoverTarget = london;
Property "faststartfailovertarget" updated
```

6. Because a far sync is being used, define the reciprocal fast-start failover target for when the `london` 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 london set property
FastStartFailoverTarget = boston;
Property "faststartfailovertarget" updated
```

7. Display the `FastStartFailoverTarget` configuration property for both the primary database and the physical standby database.

```
DGMGRL> show database boston faststartfailovertarget;
FastStartFailoverTarget = 'london'

DGMGRL> show database london FastStartFailoverTarget;
FastStartFailoverTarget = 'boston'
```

8. Modify the `ObserverReconnect` configuration property and set the value to 120 seconds.

```
DGMGRL> edit configuration set property ObserverReconnect=120;
Property "observerreconnect" updated
```

9. To enable fast-start failover when using a far sync, 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 boston redoroutes;
RedoRoutes = '(boston:bostonFS SYNC)'
DGMGRL> show database london redoroutes;
RedoRoutes = '(london:londonFS SYNC)'
```

**Note:** FASTSYNC would also be acceptable settings for the maximum availability.

10. Upgrade the protection mode to maximum availability.

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

11. 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.
```

12. Start the observer process.

```
DGMGRL> start observer
[W000 2020-06-06T23:31:17.857-04:00] FSFO target standby is
london
Observer 'host02.example.com' 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 boston.
[W000 2020-06-06T23:31:18.314-04:00] The standby london 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  
boston.
```

**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.

13. Return to the terminal session connected to host01. Simulate a crash of the primary database.

```
[oracle@host01 ~]$ pgrep -lf smon  
21521 ora_smon_boston  
[oracle@host01 ~]$ kill -9 21521  
[oracle@host01 ~]$
```

14. Return to the Observer session on host02. 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 boston  
[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.

15. Return to the terminal session on host01. Using SQL\*Plus, connect as the **SYS** user and start the primary database. Exit SQL\*Plus.

```
[oracle@host01 ~]$ 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@host01 ~]$
```

16. Launch the DGMGRL utility and connect as the **SYSDG** user.

```
[oracle@host01 ~]$ 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>@boston
```

Connected to "boston"

Connected as SYSDG.

```
DGMGRL>
```

17. Stop observer and disable Fast-Start Failover in Observe-only mode.

```
DGMGRL> stop observer
Observer stopped.
DGMGRL> DISABLE FAST_START FAILOVER
Disabled.
DGMGRL>
```

18. 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 host02 where the bostonFS far sync is currently running. After enabling fast-start failover, you will start the observer process.

### Tasks

1. Use a terminal window on host02 connected as oracle with the environment variables set to bostonFS. Launch the DGMGRL utility and connect as the SYSDG user with operating system authentication.

```
[oracle@host02 ~]$ . oraenv
ORACLE_SID = [oracle] ? bostonFS
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>@bostonFS
Connected to "bostonFS"
Connected as SYSDG.
DGMGRL>Connected as SYSDG.
```

2. 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 london

Database - london

Enterprise Manager Name: london.example.com
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):
london
```

```
Database Status:  
SUCCESS  
DGMGRL> show database london2  
  
Database - london2  
Enterprise Manager Name: london2.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):  
london2  
  
Database Status:  
SUCCESS
```

3. Display the current configuration and note the current state of fast-start failover.

```
DGMGRL> show configuration  
  
Configuration - DRSSolution  
  
Protection Mode: MaxAvailability  
Members:  
boston - Primary database  
bostonFS - Far sync instance  
london - Physical standby database  
london2 - Logical standby database  
  
Members Not Receiving Redo:  
londonFS - Far sync instance  
  
Fast-Start Failover: Disabled  
  
Configuration Status:  
SUCCESS (status updated 61 seconds ago)  
  
DGMGRL>
```

4. Enable fast-start failover.

```
DGMGRL> enable fast_start failover  
Enabled in Zero Data Loss Mode.
```

5. 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: MaxAvailability  
Lag Limit: 0 seconds  
  
Threshold: 30 seconds  
Active Target: london  
Potential Targets: "london"  
    london      valid  
Observer: (none)  
Shutdown Primary: TRUE  
Auto-reinstate: 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>
```

6. 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 host01 connected as oracle with the environment variables set to boston. Connect to the primary database using SQL\*Plus and simulate a failure by issuing the shutdown abort command. Exit SQL\*Plus.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$ 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@host01 ~]$
```

2. Observe the status output in the terminal session connected to host02 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 "london"...
[SO02 2020-06-11T16:23:56.156+00:00] Initiating Fast-start
Failover.
Performing failover NOW, please wait...
```

3. Use a terminal window on host03 connected as oracle with the environment variables set to london. Launch the DGMGRL utility and connect as the SYSDG user.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been changed from /home/oracle to
/u01/app/oracle
[oracle@host03 ~]$ 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>@london
Connected to "LONDON"
Connected as SYSDG.
```

4. 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 boston database.

```
DGMGRL> show configuration

Configuration - DRSolution

Protection Mode: MaxAvailability
Members:
london    - Primary database
londonFS - Far sync instance
```

```
boston - Physical standby database (disabled)
ORA-16661: the standby database needs to be reinstated

london2 - Logical standby database

Members Not Receiving Redo:
bostonFS - Far sync instance

Fast-Start Failover: Enabled in Zero Data Loss Mode

Configuration Status:
SUCCESS (status updated 49 seconds ago)
```

5. Use a terminal window on host01 connected as oracle with the environment variables set to boston. Connect to the former primary database (the new physical standby database) using SQL\*Plus and mount the boston database to initiate reinstatement. Exit SQL\*Plus.

```
[oracle@host01 ~]$ 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
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@host01 ~]$
```

6. Observe the output in the terminal session on host02 running the observer process.

```
2020-06-07T01:18:53.809-04:00
Initiating reinstatement for database "boston"...
Reinstating database "boston", 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 "boston" succeeded
2020-06-07T01:21:09.990-04:00
[W000 2020-06-07T01:21:09.993-04:00] Successfully reinstated
database boston.
[W000 2020-06-07T01:21:15.094-04:00] Disconnecting from database
london.
```

7. Return to the DGMGRL session running on host03 and display the configuration.

```
DGMGRL> show configuration
Configuration - DRSolution
  Protection Mode: MaxAvailability
  Databases:
    london      - Primary database
    londonFS   - Far Sync
    boston      - (*) Physical standby database
      Warning: ORA-16857: standby disconnected from redo
      source for longer than specified threshold
    london2     - Logical standby database
    bostonFS   - 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.

8. 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 boston
Database - boston

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):
    boston

Database Status:
SUCCESS

DGMGRL> show configuration

Configuration - DRSSolution

Protection Mode: MaxAvailability
Members:
london - Primary database
    londonFS - Far sync instance
    boston - (*) Physical standby database
    london2 - Logical standby database

Members Not Receiving Redo:
    bostonFS - 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 london primary database is ready for switchover.

```
DGMGRL> validate database london

Database Role:      Primary database

Ready for Switchover: Yes

Managed by Clusterware:
    london: NO
    Validating static connect identifier for the primary
    database london...
    The static connect identifier allows for a connection to
    database "london".

DGMGRL>
```

2. Validate that the boston standby database is ready for switchover.

```
DGMGRL> validate database boston

Database Role:      Physical standby database
Primary Database:   london

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

Managed by Clusterware:
    london: NO
    boston: NO
    Validating static connect identifier for the primary
    database london...
    The static connect identifier allows for a connection to
    database "london".

Log Files Cleared:
    london Standby Redo Log Files: Cleared
```

```

boston Online Redo Log Files: Not Cleared
boston Standby Redo Log Files: Available

Current Log File Groups Configuration:
  Thread #  Online Redo Log Groups  Standby Redo Log Groups
Status
                (london)          (boston)
  1            3                  2
Insufficient SRLs

Future Log File Groups Configuration:
  Thread #  Online Redo Log Groups  Standby Redo Log Groups
Status
                (boston)          (london)
  1            3                  3
Insufficient SRLs

DGMGRL>

```

3. Switch over to the `boston` database.

```

DGMGRL> switchover to boston
Performing switchover NOW, please wait...
Operation requires a connection to database "boston"
Connecting ...
Connected to "boston"
Connected as SYSDG.

New primary database "boston" is opening...
Operation requires start up of instance "london" on database
"london"
Starting instance "london"...
Connected to an idle instance.
ORACLE instance started.
Connected to "london"
Database mounted.
Database opened.
Connected to "london"
Connected to "boston"
Switchover succeeded, new primary is "boston"
DGMGRL>

```

4. 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 - DRSSolution

Protection Mode: MaxAvailability
Members:
boston      - Primary database
bostonFS    - Far sync instance
london      - (*) Physical standby database
london2     - Logical standby database

Members Not Receiving Redo:
londonFS   - Far sync instance

Fast-Start Failover: Enabled in Zero Data Loss Mode

Configuration Status:
SUCCESS      (status updated 44 seconds ago)

DGMGRL>
```

5. 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.

6. Disable fast-start failover.

```
DGMGRL> disable fast_start failover
Disabled.
```

7. Reset the protection mode back to maximum performance and exit DGMGRL.

```
DGMGRL> edit configuration set protection mode as
maxperformance;
Succeeded.
DGMGRL> exit
[oracle@host03 ~]$
```

# **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 host03 for the physical standby database and verify its usage.

### Tasks

1. Use a terminal window logged in as oracle to host03 with the environment variables set for london 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/london/rman\_change\_track.file.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been set to /u01/app/oracle
[oracle@host03 ~]$ 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/LONDON/rman_change_track.file';

Database altered.

SQL>
```

2. 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/london/rman_change_track.file
ENABLED          11599872
```

3. Exit SQL\*Plus on host03 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 em13c as the oracle OS user. Enter <password> when you are prompted for the password. Set the environment variables to rcatcdb.

```
[oracle@gateway ~]$ ssh oracle@em13c
oracle@em13c's password: <password>
[oracle@em13c ~]$ . oraenv
ORACLE_SID = [ORCL] ? rcatcdb
The Oracle base remains unchanged with value /u01/app/oracle
```

2. Invoke SQL\*Plus and connect as the SYS user with the SYSDBA privilege. List the PDBs in the connected database.

```
[oracle@em13c ~]$ 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
----- -----
2  PDB$SEED              READ ONLY NO
3  RCATPDB                READ WRITE NO
4  EMCCPDB                READ WRITE NO
```

3. Connect to the RCATPDB PDB.

```
SQL> alter session set container=rcatpdb;  
  
Session altered.  
  
SQL>
```

4. Determine where the current data files are stored at:

```
SQL> select file_name from dba_data_files;  
  
FILE_NAME  
-----  
/u01/app/oracle/oradata/RCATCDB/rcatpdb/system01.dbf  
/u01/app/oracle/oradata/RCATCDB/rcatpdb/sysaux01.dbf  
/u01/app/oracle/oradata/RCATCDB/rcatpdb/undotbs01.dbf  
/u01/app/oracle/oradata/RCATCDB/rcatpdb/users01.dbf  
  
SQL>
```

5. Create a new tablespace for the recovery manager repository using the same storage architecture as the existing files. Name the tablespace `rcts` and give it an initial size of 30MB with `autoextend` turned on.

```
SQL> create tablespace rcts datafile  
'/u01/app/oracle/oradata/RCATCDB/rcatpdb/rcts01.dbf' size 30M  
autoextend on;  
  
Tablespace created.
```

6. 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.
```

7. 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
[oracle@em13c ~]$
```

8. Use a terminal window on host01 connected as oracle with the environment variables set to boston. Launch the RMAN utility and connect to the rcatpdb service using the account just created.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$ rman catalog
rcowner/<password>@em13c:1521/rctpdbs

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
```

9. 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 host01. Connect to the boston primary database with SYSDBA privilege and register the database.

```
RMAN> connect target 'sys/<password>@boston as sysdba'

connected to target database: BOSTON (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.

2. List the DB\_UNIQUE\_NAME for all databases known to the recovery catalog.

```
RMAN> list db_unique_name of database;

List of Databases
DB Key   DB Name    DB ID        Database Role      Db_unique_name
-----  -----  -----
1       BOSTON    2732402101 PRIMARY          BOSTON
```

3. Generate a schema report for the boston primary database.

```
RMAN> report schema for db_unique_name boston;

Report of database schema for database with db_unique_name
BOSTON

List of Permanent Datafiles
=====
File Size (MB)  Tablespace          RB segs Datafile Name
-----  -----  -----
--
```

```

1    960      SYSTEM          YES
/u01/app/oracle/oradata/BOSTON/system01.dbf
3    940      SYSAUX         NO
/u01/app/oracle/oradata/BOSTON/sysaux01.dbf
4    280      UNDOTBS1        YES
/u01/app/oracle/oradata/BOSTON/undotbs01.dbf
5    270      PDB$SEED:SYSTEM  NO
/u01/app/oracle/oradata/BOSTON/pdbseed/system01.dbf
6    330      PDB$SEED:SYSAUX NO
/u01/app/oracle/oradata/BOSTON/pdbseed/sysaux01.dbf
7    5       USERS           NO
/u01/app/oracle/oradata/BOSTON/users01.dbf
8   100      PDB$SEED:UNDOTBS1 NO
/u01/app/oracle/oradata/BOSTON/pdbseed/undotbs01.dbf
9   510      DEV1:SYSTEM      YES
/u01/app/oracle/oradata/BOSTON/dev1/system01.dbf
10   370     DEV1:SYSAUX     NO
/u01/app/oracle/oradata/BOSTON/dev1/sysaux01.dbf
11   100     DEV1:UNDOTBS1   YES
/u01/app/oracle/oradata/BOSTON/dev1/undotbs01.dbf
12   5      DEV1:USERS      NO
/u01/app/oracle/oradata/BOSTON/dev1/users01.dbf

```

List of Temporary Files

File	Size (MB)	Tablespace	Maxsize (MB)	Tempfile Name
---				
1	132	TEMP	32767	
		/u01/app/oracle/oradata/BOSTON/temp01.dbf		
2	36	PDB\$SEED:TEMP	32767	
		/u01/app/oracle/oradata/BOSTON/pdbseed/temp012020-06-04_02-09-11-127-AM.dbf		
3	36	DEV1:TEMP	32767	
		/u01/app/oracle/oradata/BOSTON/dev1/temp01.dbf		

RMAN>

4. List all the archive logs for the boston primary database.

```
RMAN> list archivelog all for db_unique_name boston;

List of Archived Log Copies for database with db_unique_name
BOSTON
=====
=====

Key      Thrd Seq      S Low Time
-----
191      1    6        A 04-JUN-20
      Name:
/u01/app/oracle/fast_recovery_area/BOSTON/archivelog/2020_06_04/
o1_mf_1_6_hfjpp4gy_.arc

192      1    7        A 04-JUN-20
      Name:
/u01/app/oracle/fast_recovery_area/BOSTON/archivelog/2020_06_04/
o1_mf_1_7_hfk3hmrg_.arc

...
328      1   16        A 11-JUN-20
      Name:
/u01/app/oracle/fast_recovery_area/BOSTON/archivelog/2020_06_11/
o1_mf_1_16_hg4qd0d3_.arc

RMAN>
```

5. Display all the current configuration parameters for the boston primary database.

```
RMAN> show all for db_unique_name boston;

RMAN configuration parameters for database with db_unique_name
BOSTON 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_boston.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
```

2. 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
```

3. Configure the connect identifier for your primary database.

```
RMAN> configure db_unique_name boston connect identifier  
'boston';  
  
new RMAN configuration parameters:  
CONFIGURE DB_UNIQUE_NAME 'boston' CONNECT IDENTIFIER 'boston';  
new RMAN configuration parameters are successfully stored  
starting full resync of recovery catalog  
full resync complete
```

4. Configure the connect identifier for your physical standby database.

```
RMAN> configure db_unique_name london connect identifier  
'london';  
  
new RMAN configuration parameters:  
CONFIGURE DB_UNIQUE_NAME 'london' CONNECT IDENTIFIER 'london';  
new RMAN configuration parameters are successfully stored  
starting full resync of recovery catalog  
full resync complete
```

5. 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        BOSTON     2732402101 PRIMARY          BOSTON  
1        BOSTON     2732402101 STANDBY          LONDON
```

6. 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 london;

Report of database schema for database with db_unique_name
LONDON

List of Permanent Datafiles
=====
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

RMAN> exit;

Recovery Manager complete.
[oracle@host01 ~]$
```

## 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 exist. Use a terminal window logged in as oracle to host03. Create a symbolic link BOSTON linking to london2 so that file creation can proceed.

```
[oracle@host03 ~]$ cd /u01/app/oracle/oradata  
[oracle@host03 oradata]$ ln -s london2 BOSTON
```

**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.

2. Use a terminal window logged in as oracle to host01 with the environment variables set for boston appropriately. Launch SQL\*Plus and create a new tablespace SAMPLE in the DEV1 pluggable database with a data file /u01/app/oracle/oradata/boston/dev1/sample01.dbf and a size of 5 MB.

```
[oracle@host01 ~]$ 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/BOSTON/dev1/sample01.dbf' size 5M;
```

```
Tablespace created.
```

```
SQL>
```

3. Connect to your physical standby instance as the SYSDBA user and show the standby\_file\_management parameter.

```
SQL> connect sys/<password>@london as sysdba  
Connected.
```

```
SQL> show parameter standby_file_management
```

NAME	TYPE	VALUE
standby_file_management	string	AUTO

```
SQL>
```

4. 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$logfile;  
  
FILE# NAME  
-----  
1 /u01/app/oracle/oradata/LONDON/system01.dbf  
3 /u01/app/oracle/oradata/LONDON/sysaux01.dbf  
4 /u01/app/oracle/oradata/LONDON/undotbs01.dbf  
5 /u01/app/oracle/oradata/LONDON/pdbseed/system01.dbf  
6 /u01/app/oracle/oradata/LONDON/pdbseed/sysaux01.dbf  
7 /u01/app/oracle/oradata/LONDON/users01.dbf  
8 /u01/app/oracle/oradata/LONDON/pdbseed/undotbs01.dbf  
9 /u01/app/oracle/oradata/LONDON/dev1/system01.dbf  
10 /u01/app/oracle/oradata/LONDON/dev1/sysaux01.dbf  
11 /u01/app/oracle/oradata/LONDON/dev1/undotbs01.dbf  
12 /u01/app/oracle/oradata/LONDON/dev1/users01.dbf  
  
FILE# NAME  
-----
```

```
25 /u01/app/oracle/oradata/LONDON/dev1/sample01.dbf  
  
12 rows selected.  
  
SQL>
```

5. Connect to your logical standby instance as the `SYSDBA` user.

```
SQL> connect sys/<password>@london2 as sysdba  
Connected.
```

6. Verify that the data file has successfully been created on the logical standby database.

```
SQL> select file#,name from v$logfile;  
  
FILE# NAME  
-----  
1 /u01/app/oracle/oradata/london2/system01.dbf  
3 /u01/app/oracle/oradata/london2/sysaux01.dbf  
4 /u01/app/oracle/oradata/london2/undotbs01.dbf  
5 /u01/app/oracle/oradata/london2/pdbseed/system01.dbf  
6 /u01/app/oracle/oradata/london2/pdbseed/sysaux01.dbf  
7 /u01/app/oracle/oradata/london2/users01.dbf  
8 /u01/app/oracle/oradata/london2/pdbseed/undotbs01.dbf  
9 /u01/app/oracle/oradata/london2/dev1/system01.dbf  
10 /u01/app/oracle/oradata/london2/dev1/sysaux01.dbf  
11 /u01/app/oracle/oradata/london2/dev1/undotbs01.dbf  
12 /u01/app/oracle/oradata/london2/dev1/users01.dbf  
  
FILE# NAME  
-----  
25 /u01/app/oracle/oradata/BOSTON/dev1/sample01.dbf  
  
12 rows selected.  
  
SQL>
```

7. 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>@host01:1521/DEV1.example.com  
Connected.
```

```
SQL> create table hr.employees2 tablespace sample as select *  
from hr.employees;
```

```
Table created.
```

- Verify that the table was created by counting the number of rows it contains.

```
SQL> select count(*) from hr.employees2;  
COUNT (*)  
-----  
107
```

- Move the sample01.dbf file to sample01.sav to simulate a failure in the primary database.

```
SQL> !mv /u01/app/oracle/oradata/BOSTON/dev1/sample01.dbf  
/u01/app/oracle/oradata/BOSTON/dev1/sample01.sav
```

- Connect to the root container and shut down abort the primary database. Exit SQL\*Plus.

```
SQL> connect sys/<password>@boston 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@host01 ~]$
```

- Launch SQL\*Plus and start the database instance. Exit SQL\*Plus.

```
[oracle@host01]$ 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.
```

```

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> 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/BOSTON/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@host01 ~]$
```

12. Use RMAN to restore the missing datafile using the physical standby database over the network. Exit RMAN when done.

```

[oracle@host01 ~]$ rman target sys/<password>@boston

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: BOSTON (DBID=2732402101)

RMAN> restore datafile 25 from service 'london';

Starting restore at 07-JUN-20
using target database control file instead of recovery catalog
```

```

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 london
channel ORA_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_DISK_1: restoring datafile 00025 to
/u01/app/oracle/oradata/BOSTON/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@host01 ~]$
```

13. Launch SQL\*Plus and switch the container to the pluggable database.

```

[oracle@host01 ~]$ 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
```

CON_ID CON_NAME	OPEN MODE	RESTRICTED
2 PDB\$SEED	READ ONLY	NO
3 DEV1	MOUNTED	

SQL>  
SQL> **alter pluggable database dev1 open;**  
  
Pluggable database altered.

14. Verify that the table has been recovered and the rows exist in the table.

EMPLOYEE_ID FIRST_NAME LAST_NAME			
EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
SALARY			
SHIGGINS	515.123.8080	07-JUN-02	AC MGR
12008	101	110	
WGIETZ	515.123.8181	07-JUN-02	AC ACCOUNT
8300	205	110	

EMPLOYEE_ID FIRST_NAME LAST_NAME			
EMAIL	PHONE NUMBER	HIRE DATE	JOB ID
SALARY			
COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID	

205 Shelley Higgins  
206 William Gietz

```
-----  
107 rows selected.
```

15. 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@host01 ~]$
```

## 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 host01 as the oracle user. Make sure that you set up your environment variables correctly.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [boston] ? boston
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@host01 ~]$
```

2. Disable the redo transport service in preparation of the practice.

```
[oracle@host01 ~]$ 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>@boston
Connected to "boston"
Connected as SYSDG.
DGMGRL> edit database boston set state='TRANSPORT-OFF';
Succeeded.
DGMGRL>
```

3. Use the terminal window on host03 as the oracle user. Make sure that you set up your environment variables correctly. Launch SQL\*Plus to stop the london standby database.

```
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? london
The Oracle base has been set to /u01/app/oracle
[oracle@host03 ~]$ 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>

```

4. Open a new terminal window on host01. 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@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$ 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>

```

5. 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>

```

6. 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>

```

7. 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      bostonfs
1
/u01/app/oracle/fast_recovery_area/BOSTON/archivelog/2020_06_05/o
    1_mf_1_82_hfnrhsbz_.arc

1
/u01/app/oracle/fast_recovery_area/BOSTON/archivelog/2020_06_07/o
    1_mf_1_82_hft44916_.arc

```

```
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release
19.0.0.0.0 - Production
Version 19.3.0.0.0
[oracle@host01 ~]$
```

- 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@host01 ~]$ rm
/u01/app/oracle/fast_recovery_area/BOSTON/archivelog/2020_06_07/o1
_mf_1_82_hft44916_.arc
[oracle@host01 ~]$
```

- Return to the DGMGRL session on host01. Start the redo transport service to the physical standby database.

```
DGMGRL> edit database boston set state='TRANSPORT-ON';
Succeeded.
DGMGRL>
```

- Return to the SQL\*Plus session on host03 and start the physical standby database.

```
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> alter pluggable database dev1 open;
Pluggable database altered.
```

- 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@host03 ~]$
```

**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.

12. Return to the DGMGRL session on host01, stop the Managed Recovery Process to prepare for the standby database recovery.

```
DGMGRL> edit database london set state='APPLY-OFF';
Succeeded.
DGMGRL>
```

13. Return to the terminal session on host03. 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@host03 ~]$ 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: BOSTON (DBID=2732274290, not open)

RMAN> RECOVER STANDBY DATABASE FROM SERVICE=boston;

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 'boston';
    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 boston
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/LONDON/control01.ctl
output file
name=/u01/app/oracle/fast_recovery_area/LONDON/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 'boston';
}
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/LONDON/archivelog/2020_06_07/o1
_mf_1_60_hfr8lmg_.arc

...
File Name:
/u01/app/oracle/fast_recovery_area/LONDON/autobackup/2020_06_01
_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 boston
destination for restore of datafile 00001:
/u01/app/oracle/oradata/LONDON/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 boston
destination for restore of datafile 00003:
/u01/app/oracle/oradata/LONDON/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/LONDON/archivelog/2020_06_23/o1
_mf_1_37_hh53htbg_.arc
archived log file
name=/u01/app/oracle/fast_recovery_area/LONDON/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/LONDON/rman_change_track.file'
Executing: alter system set standby_file_management=auto
Finished recover at 23-JUN-20
RMAN> exit

Recovery Manager complete.
[oracle@host03 ~]$
```

14. Using SQL\*Plus, connect to the london database. Start the database and its PDB.

```

[oracle@host03 ~]$ 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
----- -----
    2 PDB$SEED            MOUNTED
    3 DEV1                MOUNTED

SQL> alter database open;

Database altered.

SQL> alter pluggable database dev1 open;

Pluggable database altered.

```

15. Return to the DGMGRL session on host01. Start the redo apply service.

```

DGMGRL> edit database london set state='APPLY-ON';
Succeeded.
DGMGRL>

```

16. Return to the SQL\*Plus session on host03 connected to the london 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@host03 ~]$

```

17. Return to the DGMGRL session on host01. Display the status of the data guard broker configuration.

```
DGMGRL> show configuration

Configuration - DRSolution

Protection Mode: MaxPerformance
Members:
boston - Primary database
bostonFS - Far sync instance
london - Physical standby database
london2 - Logical standby database
      Warning: ORA-16809: multiple warnings detected for the
member

Members Not Receiving Redo:
londonFS - 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 london2 standby database was caused by the missing redo from the primary database.

18. Disable the london2 logical standby database.

**Note:** We can safely disable the london2 database because it will not be used in later practices.

```
DGMGRL> disable database london2;
Disabled.
DGMGRL> show configuration

Configuration - DRSolution

Protection Mode: MaxPerformance
Members:
boston - Primary database
bostonFS - Far sync instance
london - Physical standby database
london2 - Logical standby database (disabled)
      ORA-16749: The member was disabled manually.

Members Not Receiving Redo:
```

```
londonFS - Far sync instance  
Fast-Start Failover: Disabled  
Configuration Status:  
SUCCESS      (status updated 48 seconds ago)  
DGMGRL>
```

19. 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 host01 to host03 during switchover exercises.

### Tasks

1. Use a terminal window logged in as `oracle` to host01 with the environment variables set for `boston` appropriately. Launch SQL\*Plus and set the session container to the `DEV1` pluggable database.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle

[oracle@host01 ~]$ 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.
```

2. Create and start a service with the name `PRMY.EXAMPLE.COM`.

```
SQL> exec
DBMS_SERVICE.CREATE_SERVICE('PRMY.EXAMPLE.COM', 'PRMY.EXAMPLE.COM')

PL/SQL procedure successfully completed.
```

```
SQL> exec DBMS_SERVICE.START_SERVICE('PRMY.EXAMPLE.COM')
```

```
PL/SQL procedure successfully completed.
```

3. From within SQL\*Plus, display the status of the Oracle listener running on host01 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=host01.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/host01/listener/alert/log.xml
Listening Endpoints Summary...

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=host01.example.com) (PORT=
1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps) (HOST=host01.example.com) (PORT
=5500)) (Security=(my_wallet_directory=/u01/app/oracle/admin/boston/
xdb_wallet)) (Presentation=HTTP) (Session=RAW))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps) (HOST=host01.example.com) (PORT
=5501)) (Security=(my_wallet_directory=/u01/app/oracle/admin/boston/
xdb_wallet)) (Presentation=HTTP) (Session=RAW))
Services Summary...
Service "PRMY.EXAMPLE.COM" has 1 instance(s).
  Instance "boston", status READY, has 1 handler(s) for this
service...
Service "boston.example.com" has 2 instance(s).
```

```

Instance "boston", status UNKNOWN, has 1 handler(s) for this
service...
Instance "boston", status READY, has 1 handler(s) for this
service...
Service "bostonXDB.example.com" has 1 instance(s).
Instance "boston", status READY, has 0 handler(s) for this
service...
Service "boston_DGB.example.com" has 1 instance(s).
Instance "boston", status READY, has 1 handler(s) for this
service...
Service "boston_DGMGRL.example.com" has 1 instance(s).
Instance "boston", status UNKNOWN, has 1 handler(s) for this
service...
Service "dev1.example.com" has 1 instance(s).
Instance "boston", status READY, has 1 handler(s) for this
service...
The command completed successfully

```

4. Use another terminal window logged in as oracle to host02 with the environment variables set for bostonFS appropriately. Launch SQL\*Plus and connect to the PRMY.EXAMPLE.COM service.

```

[oracle@host02 ~]$ . oranev
ORACLE_SID = [oracle] ? bostonFS
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 host01.example.com and host03.example.com, and access a service name of PRMY.EXAMPLE.COM.

5. Verify that you are indeed connected to the boston instance (the primary database).

```
SQL> select instance_name from v$instance;

INSTANCE_NAME
-----
boston
```

6. 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>
```

7. Return to the SQL\*Plus session running on `host01` 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 `host01`.

```
SQL> show con_name

CON_NAME
-----
DEV1
SQL> @/home/oracle/setup/create_trigger.sql
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
11        execute immediate 'alter pluggable database dev1 open';
12      end if;
13      dbms_service.start_service('PRMY.EXAMPLE.COM');
```

```
14    end if;
15  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@host01 ~]
```

8. Launch DGMGRL and connect to the SYSDG account. Show the configuration.

```
[oracle@host01 ~]$ dgmgrl sysdg/<password>@boston
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 "boston"
Connected as SYSDG.

DGMGRL>
DGMGRL> show configuration

Configuration - DRSolution

Protection Mode: MaxPerformance
Members:
boston - Primary database
  bostonFS - Far sync instance
  london - Physical standby database
  london2 - Logical standby database (disabled)
    ORA-16749: The member was disabled manually.

Members Not Receiving Redo:
  londonFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:
```

```
SUCCESS      (status updated 48 seconds ago)
```

```
DGMGRL>
```

9. Validate that the primary and physical standby databases are ready for switchover.

```
DGMGRL> validate database boston
```

Database Role: Primary database

Ready for Switchover: Yes

Flashback Database Status:

boston: On

london: Off

Managed by Clusterware:

boston: NO

Validating static connect identifier for the primary database  
boston...

The static connect identifier allows for a connection to  
database "boston".

```
DGMGRL> validate database london
```

Database Role: Physical standby database

Primary Database: boston

Ready for Switchover: Yes

Ready for Failover: Yes (Primary Running)

Flashback Database Status:

boston: On

london: Off

Managed by Clusterware:

boston: NO

london: NO

Validating static connect identifier for the primary database  
boston...

The static connect identifier allows for a connection to  
database "boston".

Current Log File Groups Configuration:

Thread #	Online Redo Log Groups	Standby Redo Log Groups
Status	(boston)	(london)
1	3	3
Insufficient SRLs		
Future Log File Groups Configuration:		
Thread #	Online Redo Log Groups	Standby Redo Log Groups
Status	(london)	(boston)
1	3	2
Insufficient SRLs		

10. Perform a switch over to the london physical standby database. Do not exit DGMGRL.

```
DGMGRL> switchover to london
Performing switchover NOW, please wait...
Operation requires a connection to database "london"
Connecting ...
Connected to "london"
Connected as SYSDG.
New primary database "london" is opening...
Operation requires start up of instance "boston" on database
"boston"
Starting instance "boston"...
Connected to an idle instance.
ORACLE instance started.
Connected to "boston"
Database mounted.
Database opened.
Connected to "boston"
Switchover succeeded, new primary is "london"
DGMGRL>
```

11. Return to the SQL\*Plus session on `host02`. Attempt to verify that your session is now on the `london` 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
```

12. Establish a new session using the `PRMY.EXAMPLE.COM` service.

```
SQL> connect system/<password>@prmy
Connected.
SQL>
```

13. Verify that you are now connected to the `london` instance (the primary database).

```
SQL> select instance_name from v$instance;
INSTANCE_NAME
-----
london
```

14. 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 ~]
```

15. Return to the DGMGRL session running on `host01` 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 london
Database Role: Primary database
Ready for Switchover: Yes

DGMGRL> validate database boston
Database Role: Physical standby database
Primary Database: london
Ready for Switchover: Yes
Ready for Failover: Yes (Primary Running)
Current Log File Groups Configuration:
  Thread #  Online Redo Log Groups    Standby Redo Log Groups
                (london)                  (boston)
      1            3                      2
Future Log File Groups Configuration:
  Thread #  Online Redo Log Groups    Standby Redo Log Groups
                (boston)                  (london)
      1            3                      2

DGMGRL> switchover to boston
Performing switchover NOW, please wait...
Operation requires a connection to database "boston"
Connecting ...
Connected to "boston"
Connected as SYSDG.
New primary database "boston" is opening...
Operation requires start up of instance "london" on database
"london"
Starting instance "london"...
Connected to an idle instance.
ORACLE instance started.
Connected to "london"
Database mounted.
Database opened.
Connected to "london"
Connected to "boston"
Switchover succeeded, new primary is "boston"
DGMGRL>
```

16. 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:
boston    - Primary database
bostonFS - Far sync instance
london    - Physical standby database
london2   - Logical standby database (disabled)
ORA-16749: The member was disabled manually.

Members Not Receiving Redo:
londonFS - Far sync instance

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS      (status updated 48 seconds ago)

DGMGRL>
```

17. Keep the DGMGRL session on host01 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.EXAMPLE.COM 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 host01 with the environment variables set for boston appropriately. Run the `setup18.sh` script to prepare for a simple testing.

```
[oracle@host01]$ . oraenv
ORACLE_SID = [oracle] ? boston
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$ /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@host01 ~]$
```

2. Launch SQL\*Plus and set the session container to the DEV1 pluggable database.

```
[oracle@host01 ~]$ 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.
```

3. Modify the service PRMY.EXAMPLE.COM to support Application Continuity. Exit SQL\*Plus.

```
SQL> @/home/oracle/setup/modify_svc.sql  
SQL> DECLARE  
2  params dbms_service.svc_parameter_array;  
3  BEGIN  
4  params('FAILOVER_TYPE') := 'TRANSACTION';  
5  params('REPLAY_INITIATION_TIMEOUT') := 1800;  
6  params('RETENTION_TIMEOUT') := 86400;  
7  params('FAILOVER_DELAY') := 10;  
8  params('FAILOVER_RETRIES') := 30;  
9  params('FAILOVER_RESTORE') := 'LEVEL1';  
10 params('commit_outcome') := 'true';  
11 params('aq_ha_notifications') := 'true';  
12 dbms_service.modify_service('prmy.example.com',params);  
13 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@host01 ~]$
```

**Note:** You have to set two mandatory service attributes (`FAILOVER_TYPE` and `COMMIT_OUTCOME`) to enable the Application Continuity feature.

4. Examine the script that you will soon use to execute the practice application. The script executes the application code called `actest.jar`.

```
[oracle@host01 ~]$ 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@host01 ~]$
```

5. Examine the properties file.

- Use the Replay Data source: oracle.jdbc.replay.OracleDataSourceImpl
- Use the Application Continuity enabled service: prmy.example.com
- Use the Application Continuity supported connection pool: UCP

```
[oracle@host01 ~]$  
cat /home/oracle/setup/lab_18-2/actest_replay.properties  
  
username=hr  
password=oracle_4U  
autoCommit=false  
  
# Use new replay datasource  
datasource=oracle.jdbc.replay.OracleDataSourceImpl  
  
url=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=host01.example.com) (PORT=1521) (ADDRESS=(PROTOCOL=tcp) (HOST=host03.example.com) (PORT=1521)) (CONNECT_DATA=(SERVICE_NAME=prmy.example.com)))  
  
# 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@host01 ~]$
```

6. 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=(PROTOCOL=tcp) (HOST=host01.example.com) (PORT=1521)) (CONNECT_DATA=(SERVICE_NAME=PRMY.EXAMPLE.COM))) (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=host03.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
...
...
```

7. While the application continues to execute, return to the DGMGRL session on host01. 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:
boston - Primary database
bostonFS - Far sync instance
london - Physical standby database
london2 - Logical standby database (disabled)
ORA-16749: The member was disabled manually.

Members Not Receiving Redo:
londonFS - Far sync instance

Fast-Start Failover: Disabled
```

```
Configuration Status:  
SUCCESS      (status updated 48 seconds ago)  
  
DGMGRL>
```

8. Verify that both databases are ready for switchover, and then perform a switchover.

```
DGMGRL> validate database boston  
  
Database Role:      Primary database  
  
Ready for Switchover: Yes  
....  
  
DGMGRL> validate database london  
  
Database Role:      Physical standby database  
Primary Database:   boston  
  
Ready for Switchover: Yes  
Ready for Failover:   Yes (Primary Running)  
....  
  
DGMGRL> switchover to london  
Performing switchover NOW, please wait...  
Operation requires a connection to database "london"  
Connecting ...  
Connected to "london"  
Connected as SYSDG.  
New primary database "london" is opening...  
Operation requires start up of instance "boston" on database  
"boston"  
Starting instance "boston"...  
Connected to an idle instance.  
ORACLE instance started.  
Connected to "boston"  
Database mounted.  
Database opened.  
Connected to "boston"  
Switchover succeeded, new primary is "london"  
DGMGRL>
```

9. Return to the terminal session on host02. Press **Ctrl+C** to abort the application.

```
...
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 active connections, avg response time from db 111 r
2 active connections, avg response time from db 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
...
^C[oracle@host02 lab_18-2]$
```

**Note:** Your application is still connected without having any connection issues. There was only a small delay during the switchover operation.

10. **(Optional)** Return to the DGMGRL session on host01. Optionally, switch back to the original state.

```
DGMGRL> switchover to boston
Performing switchover NOW, please wait...
Operation requires a connection to database "boston"
Connecting ...
Connected to "boston"
Connected as SYSDG.
New primary database "boston" is opening...
Operation requires start up of instance "london" on database
"london"
Starting instance "london"...
Connected to an idle instance.
ORACLE instance started.
Connected to "london"
Database mounted.
Database opened.
Connected to "london"
Connected to "boston"
```

```
Switchover succeeded, new primary is "boston"  
DGMGRL>
```

11. Exit DGMGRL on host01.



# **Practices for Lesson 19: Patching and Upgrading Databases in a Data Guard Configuration**

## **Practices for Lesson 19**

---

There are no practices for this lesson.