ORACLE UNIVERSITY

Oracle Database 12*c*: RAC Administration

Activity Guide
D81250GC10
Edition 1.0 | September 2014 | D85495

Learn more from Oracle University at **oracle.com/education/**

ORACLE

Authors

Jim Womack, Dominique Jeunot

Technical Contributors and Reviewers

Allan Graves, Gerlinde Frenzen, Branislav Valny, Herbert Bradbury, Ira Singer, Harald Van Breederode, Joel Goodman, Sean Kim, Andy Fortunak, Al Flournoy, Markus Michalewicz, Maria BIllings, Mark Scardina, Ron Soltani

This book was published using: Oracle Tutor

Table of Contents

Practices for Lesson 1: Grid Infrastructure Overview

Chapter 1

Practice 1-1: Configuring a Flex Cluster with Flex ASM

Overview

In this practice, you will install and configure a new Flex Cluster with Flex ASM. You will install it to three nodes: `host01`, `host02`, and `host03`. You will designate `host01`, `host02`, and `host03` to be hub nodes.

1. As the `root` user, shut down `host04` and `host05` from your desktop. Wait for a few moments and confirm with the `xm list` command. Next, set the time across all nodes using the command shown below. Then restart the NAMED and NTPD services to ensure viability and availability of the services for the software installation.

```
[vncuser@classroom_pc ~]$ su -

[root@classroom_pc ~]$ xm shutdown host04
[root@classroom_pc ~]$ xm shutdown host05

[root@classroom_pc ~]# xm list
Name                        ID   Mem VCPUs      State   Time(s)
Domain-0                     0  1124     2     r----- 272227.7
host01                      61  4200     1     -b----  38968.4
host02                      67  3200     2     -b----     13.4
host03                      66  3200     2     -b----    162.5

[root@classroom_pc ~]# TIME="`date +%T`";for H in host01 host02
host03;do ssh $H date -s $TIME;done

[root@classroom_pc ~]# service ntpd restart
Shutting down ntpd:                                        [FAILED]
ntpd: Synchronizing with time server:                      [  OK  ]
Starting ntpd:                                             [  OK  ]

[root@classroom_pc ~]# service named restart
Stopping named: .                                          [  OK  ]
Starting named:
```

2. Open an ssh session as `root` to host01. Start the local naming cache daemon on all three cluster nodes with the `service nscd start` command. To make sure `nscd` starts at reboot, execute the `chkconfig nscd command`. **Perform these steps on all three of your nodes**.

```
[root@classroom_pc ~]# ssh host01
root@host01's password:

[root@host01 ~]# service nscd start
Starting nscd: [  OK  ]
[root@host01 ~]# chkconfig nscd on

[root@host01 ~]# ssh host02 service nscd start
Starting nscd: [  OK  ]
[root@host01 ~]# ssh host02 chkconfig nscd on

[root@host01 ~]# ssh host03 service nscd start
Starting nscd: [  OK  ]
[root@host01 ~]# ssh host03 chkconfig nscd on
```

3. As the `root` user, run the `/stage/RAC/labs/less_01/limits.sh` script on `host01`. This script replaces the profile for the `oracle` and `grid` users and replaces `/etc/profile`. It replaces the `/etc/security/limits.conf` file with a new one with entries for `oracle` and `grid`. It also installs the CVU rpm.

```
[root@host01 ~]# cat /stage/RAC/labs/less_01/bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin
export PATH

umask 022


[root@host01 ~]# cat /stage/RAC/labs/less_01/profile
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

pathmunge () {
        if ! echo $PATH | /bin/egrep -q "(^|:)$1($|:)" ; then
           if [ "$2" = "after" ] ; then
              PATH=$PATH:$1
           else
```

```
                PATH=$1:$PATH
            fi
        fi
}

# ksh workaround
if [ -z "$EUID" -a -x /usr/bin/id ]; then
        EUID=`id -u`
        UID=`id -ru`
fi

# Path manipulation
if [ "$EUID" = "0" ]; then
        pathmunge /sbin
        pathmunge /usr/sbin
        pathmunge /usr/local/sbin
fi

# No core files by default
ulimit -S -c 0 > /dev/null 2>&1

if [ -x /usr/bin/id ]; then
        USER="`id -un`"
        LOGNAME=$USER
        MAIL="/var/spool/mail/$USER"
fi

HOSTNAME=`/bin/hostname`
HISTSIZE=1000

if [ -z "$INPUTRC" -a ! -f "$HOME/.inputrc" ]; then
    INPUTRC=/etc/inputrc
fi

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC

for i in /etc/profile.d/*.sh ; do
    if [ -r "$i" ]; then
        . $i
    fi
done

if [ $USER = "oracle" ] || [ $USER = "grid" ]; then
        umask 022
        if [ $SHELL = "/bin/ksh" ]; then
             ulimit -p 16384
             ulimit -n 65536
        else
             ulimit -u 16384 -n 65536
        fi
fi
```

```
unset i
unset pathmunge


[root@host01 ~]# cat /stage/RAC/labs/less_01/limits.conf

#        - priority - the priority to run user process with
#        - locks - max number of file locks the user can hold
#        - sigpending - max number of pending signals
#        - msgqueue - max memory used by POSIX message queues
(bytes)
#        - nice - max nice priority allowed to raise to
#        - rtprio - max realtime priority
#<domain>      <type>  <item>         <value>

#*               soft    core            0
#*               hard    rss             10000
#@student        hard    nproc           20
#@faculty        soft    nproc           20
#@faculty        hard    nproc           50
#ftp             hard    nproc           0
#@student        -       maxlogins       4
# End of file
oracle  soft  nofile     131072
oracle  hard  nofile     131072
oracle  soft  nproc 131072
oracle  hard  nproc 131072
oracle  soft  core  unlimited
oracle  hard  core  unlimited
oracle  soft  memlock    3500000
oracle  hard  memlock    3500000
grid    soft    nofile  131072
grid    hard    nofile  131072
grid    soft    nproc   131072
grid    hard    nproc   131072
grid    soft    core    unlimited
grid    hard    core    unlimited
grid    soft    memlock 3500000
grid    hard    memlock 3500000
# Recommended stack hard limit 32MB for oracle installations
# oracle   hard   stack    32768


[root@host01 ~]# /stage/RAC/labs/less_01/limits.sh
...

[root@host01 ~]#
```

4. Create the installation directories for `grid` and `oracle`-owned software. Set the ownership and permissions of the directories using the `/stage/RAC/labs/less_01/cr_dir.sh` script.

```
[root@host01 ~]# cat /stage/RAC/labs/less_01/cr_dir.sh

#!/bin/bash

mkdir -p /u01/app/12.1.0/grid
mkdir -p /u01/app/grid
mkdir -p /u01/app/oracle
chown -R grid:oinstall /u01
chown oracle:oinstall /u01/app/oracle
chmod -R 775 /u01/

ssh host02 mkdir -p /u01/app/12.1.0/grid
ssh host02 mkdir -p /u01/app/grid
ssh host02 mkdir -p /u01/app/oracle
ssh host02 chown -R grid:oinstall /u01
ssh host02 chown oracle:oinstall /u01/app/oracle
ssh host02 chmod -R 775 /u01/

ssh host03 mkdir -p /u01/app/12.1.0/grid
ssh host03 mkdir -p /u01/app/grid
ssh host03 mkdir -p /u01/app/oracle
ssh host03 chown -R grid:oinstall /u01
ssh host03 chown oracle:oinstall /u01/app/oracle
ssh host03 chmod -R 775 /u01/

ssh host05 chmod -R 775 /u01/

[root@host01 ~]# /stage/RAC/labs/less_01/cr_dir.sh
...
[root@host01 ~]#
```

5. Next, establish a terminal session connected to `host01` as the `grid` OS user. Ensure that you specify the **`-X`** option for `ssh` to configure the ***`X`*** environment properly for the `grid` user.

```
[root@classroom_pc ~]$ ssh -X grid@host01
grid@host01's password:
[grid@host01 ~]$
```

6. Start the Oracle Clusterware release 12.1 installer. When the installer is displayed, click Cancel to exit the installer (this addresses a Java bug that crops up occasionally). Re-start the installer..

```
[grid@host01 ~]$ /stage/clusterware/runInstaller
Starting Oracle Universal Installer...

#### Exit the installer by clicking Cancel, and then Yes to exit,
then restart the installer ####

[grid@host01 ~]$ /stage/clusterware/runInstaller
Starting Oracle Universal Installer...
```

7. On the Download Software Updates screen, click Next to accept the default selection (Skip software updates).

8. On the Select Installation Option screen, click Next to accept the default selection (Install and Configure Oracle Grid Infrastructure for a Cluster).

9. On the Select Cluster Type screen, select Configure a Flex Cluster and click Next.

10. On the Select Product Languages screen, click Next to accept the default selection (English).

11. Use the Grid Plug and Play Information screen to configure the following settings:
    - Cluster Name: `cluster01`
    - SCAN Name: `cluster01-scan.cluster01.example.com`
    - SCAN Port: `1521`
    - GNS VIP Address: `192.0.2.155`
    - GNS Sub Domain: `cluster01.example.com`

    Make sure that the “Configure GNS” and “Configure nodes Virtual IPs…” check boxes are selected. Click on the “create a new GNS” radio button and then click Next.

12. On the Cluster Node Information screen, click Add to begin the process of specifying additional cluster nodes.

    Click the Add button and add `host02.example.com`. Make sure to set Node Role to **`HUB`**, and click OK. Click the add button again and add `host03.example.com`. Make sure the Node Role is set to **`HUB`** and click OK.

13. Click the SSH Connectivity button. Enter `oracle` into the OS Password field and click Test to confirm that the required SSH connectivity is configured across the cluster. Your lab environment is preconfigured with the required SSH connectivity so you will next see a dialog confirming this. Click OK to continue. Review the information in the Cluster Node Information page and click Next.

14. On the Specify Network Interface Usage screen, ensure that network interface `eth0` is designated as the Public network and that network interface `eth1` is designated as the ASM & Private network. The eth2 interface should be designated “ASM & Private network”. Click Next to continue.

15. On the Grid Infrastructure Management Repository screen, select No, dismiss the warning pop-up by clicking Yes, and click Next to continue.

16. On the Create ASM Disk Group screen, click Change Discovery Path to customize the ASM Disk Discovery Path.

17. In the Change Disk Discovery Path dialog box, set the Disk Discovery Path to `/dev/asmdisk*` and click OK.

18. On the Create ASM Disk Group screen, make sure the Disk group name is DATA and select the first 10 candidate disks in the list:
    - `/dev/asmdisk1p1`
    - `/dev/asmdisk1p10`
    - `/dev/asmdisk1p11`
    - `/dev/asmdisk1p12`
    - `/dev/asmdisk1p2`
    - `/dev/asmdisk1p3`
    - `/dev/asmdisk1p4`
    - `/dev/asmdisk1p5`
    - `/dev/asmdisk1p6`
    - `/dev/asmdisk1p7`

    Select Normal for the Redundancy and click Next to continue.

19. On the Specify ASM Password screen, select ‘Use same passwords for these accounts’ and enter `oracle_4U` as the password. Then click Next to continue.

20. On the Failure Isolation Support screen, click Next to accept the default setting (Do not use IPMI).

21. On the Privileged Operating System Groups, the values should default to the following:
    - Oracle ASM Administrator Group: `asmadmin`
    - Oracle ASM DBA Group: `asmdba`
    - Oracle ASM Operator Group : `asmoper`

    Click Next to accept the default values.

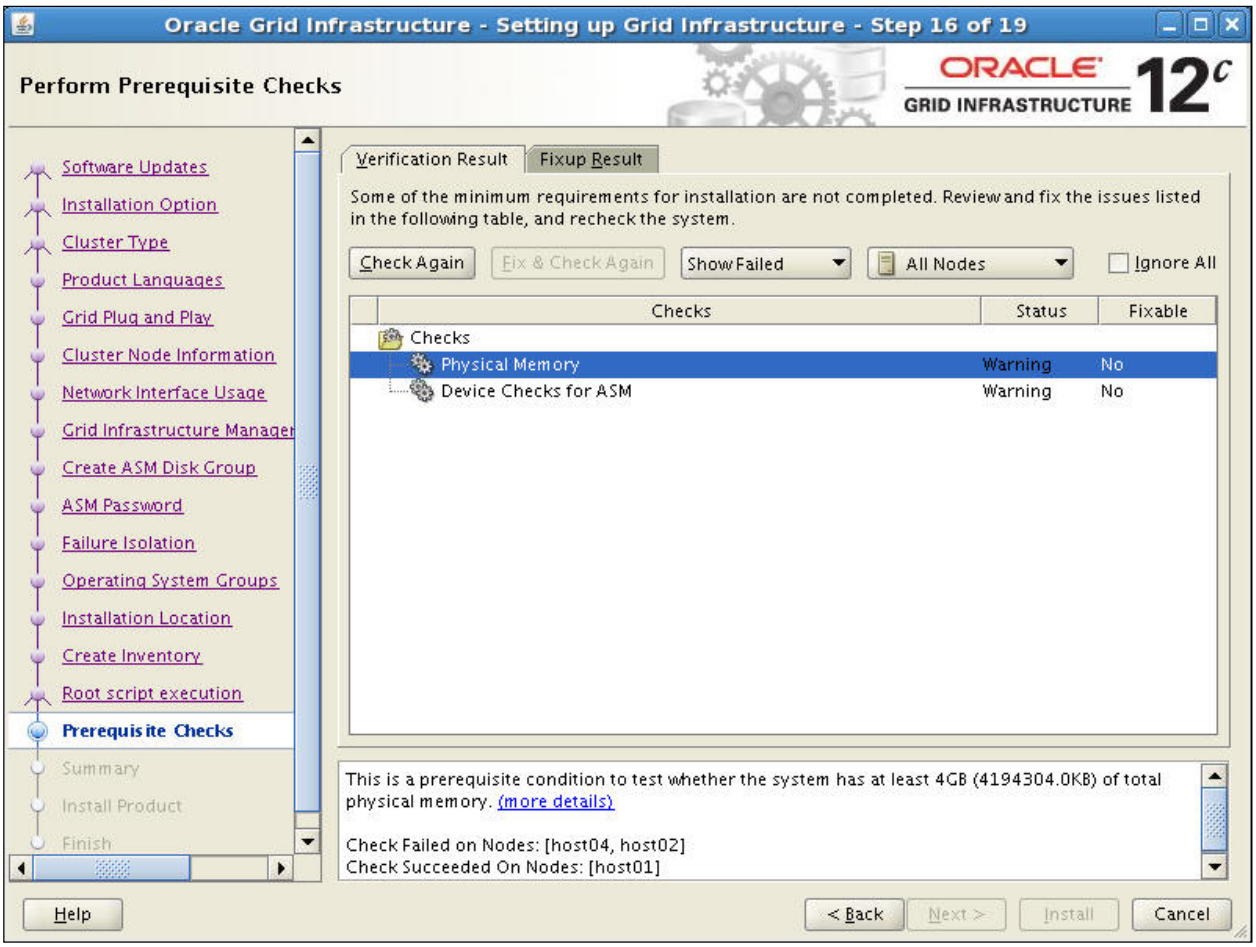
22. On the Specify Installation Location screen, change the value of the Default Software Location to **`/u01/app/12.1.0/grid`**.

23. On the Create Inventory screen, click Next to accept the default installation inventory location of `/u01/app/oraInventory`.

24. On the Root script execution configuration screen, check 'Automatically run configuration scripts' and select 'Use "`root`" user credential'. Enter `oracle` as the password and click Next to proceed.

25. Wait while a series of prerequisite checks are performed.

26. Because of the constraints of this lab environment, you may see a series of warnings resulting from the prerequisite checks. If the warnings relate to 'Physical Memory' and 'Device Checks for ASM' (as illustrated in the following screenshot) you may safely ignore them. Check 'Ignore all' and then click Next to continue.



27. In the confirmation dialog, click Yes to ignore the prerequisites flagged by the installer.

28. Examine the Summary screen. When ready, click Install to begin the installation. Oracle Grid Infrastructure release 12.1 will now install on the cluster. The Install Product screen follows the course of the installation.

29. Oracle Universal Installer pauses prior to executing the root configuration scripts. Click Yes to proceed. Oracle Universal Installer now automatically executes the root configuration scripts and you can follow the progress using the Install Product screen.

30. After configuration completes you will see the following message:
"The installation of Oracle Grid Infrastructure for a Cluster was successful"
Click Close to close Oracle Universal Installer.

31. Back in your terminal session, configure the environment using the `oraenv` script. Enter `+ASM1` when you are prompted for an `ORACLE_SID` value.

```
[grid@host01 ~]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base has been set to /u01/app/grid
[grid@host01 ~]$
```

32. Now check the status of the cluster. Ensure that all the listed services are online on all the cluster nodes.

```
[grid@host01 ~]$ crsctl check cluster -all
**************************************************************
host01:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
host02:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
host03:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
[grid@host01 ~]$
```

33. List the Clusterware resources. Ensure that all the Clusterware resources are running as shown in the following output. Notice the following new or altered resources in release 12.1:
    - Flex ASM listeners: `ora.ASMNET`*n*`LSNR_ASM.lsnr`
    - Flex ASM ADVM Proxy instances: `ora.proxy_advm`
    - Flex ASM instances: `ora.asm`

```
[grid@host01 ~]$ crsctl status resource -t
--------------------------------------------------------------------------------
Name           Target  State        Server                   State details
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.ASMNET1LSNR_ASM.lsnr
               ONLINE  ONLINE       host01                   STABLE
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
ora.ASMNET2LSNR_ASM.lsnr
               ONLINE  ONLINE       host01                   STABLE
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
ora.DATA.dg
               ONLINE  ONLINE       host01                   STABLE
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
ora.LISTENER.lsnr
               ONLINE  ONLINE       host01                   STABLE
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
ora.net1.network
               ONLINE  ONLINE       host01                   STABLE
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
ora.ons
               ONLINE  ONLINE       host01                   STABLE
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
ora.proxy_advm
               ONLINE  ONLINE       host01                   STABLE
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
```

```
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       host02                   STABLE
ora.LISTENER_SCAN2.lsnr
      1        ONLINE  ONLINE       host01                   STABLE
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       host03                   STABLE
ora.asm
      1        ONLINE  ONLINE       host01                   STABLE
      2        ONLINE  ONLINE       host02                   STABLE
      3        ONLINE  ONLINE       host03                   STABLE
ora.cvu
      1        ONLINE  ONLINE       host01                   STABLE
ora.gns
      1        ONLINE  ONLINE       host01                   STABLE
ora.gns.vip
      1        ONLINE  ONLINE       host01                   STABLE
ora.host01.vip
      1        ONLINE  ONLINE       host01                   STABLE
ora.host02.vip
      1        ONLINE  ONLINE       host02                   STABLE
ora.host03.vip
      1        ONLINE  ONLINE       host03                   STABLE
ora.oc4j
      1        ONLINE  ONLINE       host01                   STABLE
ora.scan1.vip
      1        ONLINE  ONLINE       host02                   STABLE
ora.scan2.vip
      1        ONLINE  ONLINE       host01                   STABLE
ora.scan3.vip
      1        ONLINE  ONLINE       host03                   STABLE
--------------------------------------------------------------------------------
[grid@host01 ~]$
```

34. At this point you have configured a Flex Cluster with Flex ASM. Next, you will install database software and create a RAC database on the cluster. In preparation for this you will now create another ASM disk group to host the Fast Recovery Area (FRA). Start the ASM Configuration Assistant (`asmca`).

```
[grid@host01 ~]$ asmca
```

35. After the ASM Configuration Assistant appears, click Create.

36. In the Create Disk Group window, enter `FRA` as the disk group name and select first three candidate disks (`/dev/asmdisk1p8, /dev/asmdisk1p9` and `/dev/asmdisk2p1`). Make sure the Redundancy is ***External***. Then click OK to create the disk group.

37. After the disk group creation process completes you will see a dialog window indicating that the disk group has been created. Click OK to proceed.

38. Click Exit to quit the ASM Configuration Assistant.

39. Click Yes to confirm that you want to quit the ASM Configuration Assistant.

40. Close all terminal windows opened for these practices.

Practices for Lesson 2: RAC Databases Overview & Architecture

Chapter 2

Practices for Lesson 2

Practices Overview

There are no practices for this lesson.

Practices for Lesson 3: Installing and Configuring Oracle RAC

Chapter 3

Practice 3-1: Installing RAC Database Software and Creating a RAC Database

Overview

In this practice you will install Oracle Database 12*c* software and create an Oracle RAC database.

1. Establish an `ssh` connection using the `-X` option as the `oracle` user.

```
[vncuser@classsroom_pc ~]$ ssh -X oracle@host01
oracle@host01's password:
[oracle@host01 ~]$
```

2. Change directory to `/stage/database/` and start the installer.

```
[oracle@host01 ~]$ cd /stage/database
[oracle@host01 Disk1]$ ./runInstaller
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 500 MB.   Actual 7934 MB
Passed
Checking swap space: must be greater than 150 MB.   Actual 8632 MB
Passed
Checking monitor: must be configured to display at least 256
colors.    Actual 16777216    Passed
Preparing to launch Oracle Universal Installer from
/tmp/OraInstall2013-05-22_03-35-08PM. Please wait ...
```

3. On the Confirm Security Updates screen, deselect “I wish to receive security updates” and click Next. Click Yes to confirm that you will not configure security updates for this installation.

4. On the Download Software Updates screen, click Next to accept the default selection (Skip software updates).

5. On the Select Installation Option screen, select Install database software only and click Next.

6. On the Gird Installation Options screen, click Next to accept the default selection (Oracle Real Application Clusters database installation).

7. On the Select List of Nodes screen, select `host01`, `host02`, and `host03` and click SSH Connectivity.

8. Enter `oracle` into the OS Password field and click Test to confirm that the required SSH connectivity is configured across the cluster. Your laboratory environment is preconfigured with the required SSH connectivity so you will next see a dialog confirming this. Click OK to continue.

9. If the required SSH connectivity was not present you could now click Setup to perform the required configuration. However, since the laboratory environment is already configured correctly, click Next to continue.

10. On the Select Product Languages screen, click Next to accept the default selection (English).

11. On the Select Database Edition screen, click Next to accept the default selection (Enterprise Edition).

12. On the Specify Installation Location screen, click Next to accept the default installation location. The Oracle base should be `/u01/app/oracle` and the Software location should be `/u01/app/oracle/product/12.1.0/dbhome_1`.

13. On the Privileged Operating System Groups screen, click Next to accept the default settings. They should all be `dba` except Database Operator which should be `oper`.

14. On the System Prerequisite Checks page, a series of prerequisite checks is performed.

15. Start Examine the Summary screen. When ready, click Install to start the installation.

16. Oracle Database release 12.1 software now installs on the cluster. The Install Product screen follows the course of the installation.

17. Near the end of the installation process, you will see the Execute Configuration scripts dialog box. Back in your `root` terminal session on `host01`, execute the configuration script. Press the Enter key when you are prompted for the local bin directory location. Run the script on `host02` and `host03` as shown below.

```
[root@host01 ~]# /u01/app/oracle/product/12.1.0/dbhome_1/root.sh
Performing root user operation for Oracle 12c

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/product/12.1.0/dbhome_1

Enter the full pathname of the local bin directory:
[/usr/local/bin]:
The contents of "dbhome" have not changed. No need to overwrite.
The contents of "oraenv" have not changed. No need to overwrite.
The contents of "coraenv" have not changed. No need to overwrite.

Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root script.
Now product-specific root actions will be performed.
You have new mail in /var/spool/mail/root

[root@host01 ~]# ssh host02
/u01/app/oracle/product/12.1.0/dbhome_1/root.sh
root@host02's password:
Performing root user operation for Oracle 12c

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/product/12.1.0/dbhome_1

Enter the full pathname of the local bin directory:
[/usr/local/bin]:
The contents of "dbhome" have not changed. No need to overwrite.
The contents of "oraenv" have not changed. No need to overwrite.
The contents of "coraenv" have not changed. No need to overwrite.

Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root script.
Now product-specific root actions will be performed.
```

```
[root@host01 ~]# ssh host03
/u01/app/oracle/product/12.1.0/dbhome_1/root.sh
root@host03's password:
Performing root user operation for Oracle 12c

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/product/12.1.0/dbhome_1

Enter the full pathname of the local bin directory:
[/usr/local/bin]:
The contents of "dbhome" have not changed. No need to overwrite.
The contents of "oraenv" have not changed. No need to overwrite.
The contents of "coraenv" have not changed. No need to overwrite.

Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root script.
Now product-specific root actions will be performed.

[root@host01 ~]#
```

18. After you have executed the required configuration script on your cluster nodes, return to your Oracle Universal Installer session and click OK to proceed.

19. After configuration completes you will see the Finish screen. Click Close to close Oracle Universal Installer.

Practice 3-2: Creating a RAC Database

Overview

In this practice you will create an Oracle RAC database.

1. Using the same oracle terminal session from the previous practice, change directory to `/u01/app/oracle/product/12.1.0/dbhome_1/bin/` and execute `dbca`.

```
[oracle@host01 ~]$ cd /u01/app/oracle/product/12.1.0/dbhome_1/bin
[oracle@host01 bin]$ ./dbca
```

2. On the Database Operation screen, click Next to accept the default selection (Create Database).

3. On the Creation Mode screen, click Next to accept the default selection (Advanced Mode).

4. On the Database Template screen, click Next to accept the default settings for a Policy-Managed RAC Database using the General Purpose template).

5. On the Database Identification screen, specify `orcl` as the Global Database Name and click Next.

6. On the Database Placement screen, specify **`orcldb`** for the Server pool Name and set its cardinality to **3**. Click Next to proceed.

7. On the Management Options screen, click Next to accept the default selections (Configure EM Database Express and Run CVU Checks Periodically).
8. On the Database Credentials screen, select 'Use the Same Administrative Passwords for All Accounts' and enter `oracle_4U` as the password. Then click Next to continue.

9. On the Storage Locations screen, make the following adjustments:
   - Database File Locations: +DATA
   - Fast Recovery Area: +FRA
   - Fast Recovery Area Size: 5400

   Click Next to continue.

10. On the Database Options screen, select Sample Schemas and click Next.

11. On the Memory tab, in the Initialization Parameters screen, set Memory Size (SGA and PGA) to 1300 and click on the Character Sets tab.

12. On the Character Sets tab, in the Initialization Parameters screen, select Use Unicode (AL32UTF8) and click Next.

13. On the Creation Options screen, click Next to accept the default selection (Create Database).

14. Wait while a series of prerequisite checks are performed.

15. Examine the Summary screen. When you are ready, click Finish to start the database creation process.

16. Follow the database creation process on the Progress Page.

17. Examine the dialog which indicates that the database creation process is completed. Take note of the EM Database Express URL. When you are ready, click Exit continue.

18. Click Close to quit the Database Configuration Assistant.

19. Back in the `oracle` user terminal, configure the environment using the `oraenv` script. Enter `orcl` when you are prompted for an `ORACLE_SID` value.

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

20. Use the `srvctl` command to check on which cluster nodes the database instances are running.

```
[oracle@host01 bin]$ srvctl status database -db orcl
Instance orcl_1 is running on node host03
Instance orcl_2 is running on node host01
Instance orcl_3 is running on node host02
[oracle@host01 bin]$
```

21. Close all terminal windows opened for these practices.

Congratulations! You have successfully configured an Oracle Database 12*c* Flex Cluster with Flex ASM and a RAC database.

Practices for Lesson 4: Oracle RAC Administration

Chapter 4

Practices for Lesson 4: Overview

Practices Overview

In these practices, you will contrast operating system, password file authenticated connections, and Oracle database authenticated connections. You will also learn to stop a complete ORACLE_HOME component stack.

Practice 4-1: Operating System and Password File Authenticated Connections

Overview

In this practice, you adjust initialization parameters in the SPFILE, and stop and start the ASM instances on local and remote nodes.

1. Connect to your first node as the `oracle` user and set up your environment variables by using the `oraenv` script.

```
[vnctech@classroom_pc ~]$ ssh oracle@host01
Password: <oracle>
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$$
```

2. Identify all the database instance names that are currently executing on your machine by using the Linux `ps` command.

   Note: All database instances have a mandatory background process named pmon, and the instance name will be part of the complete process name.

```
[oracle@host01 ~]$ ps -ef | grep -i pmon
grid      3529     1  0 06:45 ?        00:00:16 asm_pmon_+ASM1
grid      8669     1  0 06:50 ?        00:00:10 apx_pmon_+APX1
oracle   15813     1  0 08:02 ?        00:00:18 ora_pmon_orcl_3
oracle   19607 16483  0 15:24 pts/1    00:00:00 grep -i pmon
```

3. Attempt to make a local connection to the `orcl_n` instance by using SQL*Plus with the `sysdba` privilege. This is known as operating system authentication because a password is not needed. What happens when you are trying to connect to the instance?

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

SQL*Plus: Release 12.1.0.1.0 Production on Wed Sep 11 15:25:43
2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> exit
Disconnected
[oracle@host01 ~]$
```

4. Attempt to connect to the instance by using a network connection string `@orcl` with the `sysdba` privilege. This is known as password file authentication. Is the connection successful this time?

```
[oracle@host01 ~]$ sqlplus sys@orcl as sysdba

SQL*Plus: Release 12.1.0.1.0 Production on Wed Sep 11 15:27:50
2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Enter password: oracle_4U << Password is not displayed

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
[oracle@host01 ~]$
```

5. Display the values of the environment variables (`ORACLE_BASE`, `ORACLE_HOME`, `ORACLE_SID`, and so on) that were defined with the `oraenv` script in step 1.

```
[oracle@host01 ~]$ env | grep ORA
ORACLE_SID=orcl
ORACLE_BASE=/u01/app/oracle
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
[oracle@host01 ~]$
```

6. Modify the `ORACE_SID` environment variable to match the actual database instance name for the `orcl` database.

```
[oracle@host01 ~]$ export ORACLE_SID=orcl_3
[oracle@host01 ~]$
```

7. Attempt the local connection with system authentication to the local instance by using SQL*Plus with the `sysdba` privilege. This is the same command as in step 3.

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

```
SQL*Plus: Release 12.1.0.1.0 Production on Wed Sep 11 15:35:32
2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.



Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

SQL>
```

8. Query the `instance_name` column of `the v$instance` dynamic performance view to validate the instance that you connected with. Exit SQL*Plus when finished.

```
SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
orcl_3

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
[oracle@host01 ~]$
```

Practice 4-2: Oracle Database Authenticated Connections

Overview

In this practice, you will make multiple Oracle database authenticated connections to a database instance and notice the effects of load-balanced connections.

1. From your first node, connected as the `oracle` user, validate the instance names on each host.

```
[oracle@host01 ~]$ srvctl status database -d orcl
Instance orcl_1 is running on node host02
Instance orcl_2 is running on node host03
Instance orcl_3 is running on node host01
[oracle@host01 ~]$
```

2. Connect to a database instance by using SQL*Plus with the system account. This is known as Oracle database authentication. After it is connected, query the `instance_name` column from the `v$instance` dynamic performance view.
   Note: Your instance names may vary from the ones displayed below:

```
[oracle@host01 ~]$ sqlplus system@orcl

SQL*Plus: Release 12.1.0.1.0 Production on Wed Sep 11 16:02:01
2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Enter password: oracle_4U << Password is not displayed

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
orcl_2

SQL>
```

3. Use the SQL*Plus host command to temporarily exit SQL*Plus and return to the operating system prompt.
   Note: SQL*Plus is still running when this is performed. Repeat the previous step from the operating system prompt to establish a third SQL*Plus session and database instance connection. What instance name did you connect to?

```
SQL> !
[oracle@host01 ~]$ sqlplus system@orcl

SQL*Plus: Release 12.1.0.1.0 Production on Wed Sep 11 16:07:15
2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Enter password: oracle_4U << Password is not displayed
Last Successful login time: Wed Sep 11 2013 16:02:32 +00:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
orcl_1

SQL>
```

4. Use the SQL*Plus host command to temporarily exit SQL*Plus and return to the operating system prompt. Note: SQL*Plus is still running when this is performed. Validate that you are still on your first node. Repeat the previous step from the operating system prompt to establish a third SQL*Plus session and database instance connection. What instance name did you connect to?

```
SQL> !
[oracle@host01 ~]$ sqlplus system@orcl

SQL*Plus: Release 12.1.0.1.0 Production on Wed Sep 11 16:07:15
2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.
```

```
Enter password: oracle_4U << Password is not displayed
Last Successful login time: Wed Sep 11 2013 16:02:32 +00:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
orcl_3

SQL>
```

5. Exit the three SQL*Plus sessions that are currently executing on the first node.

```
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
[oracle@host01 ~]$ exit
exit

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
[oracle@host01 ~]$ exit
exit

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
[oracle@host01 ~]$
```

Practice 4-3: Stopping a Complete ORACLE_HOME Component Stack

Overview

In this practice, you will use the `srvctl` utility to stop all resource components executing from a single Oracle home location.

1. Validate that the instances are running on each node of the cluster using the `ps` command.

```
[oracle@host01 ~]$ ps -ef|grep -i pmon
grid      3529     1  0 06:45 ?        00:00:18 asm_pmon_+ASM1
grid      8669     1  0 06:50 ?        00:00:11 apx_pmon_+APX1
oracle   15813     1  0 08:02 ?        00:00:20 ora_pmon_orcl_3
oracle   24700 16483  0 16:25 pts/1    00:00:00 grep -i pmon
[oracle@host01 ~]$ ssh host02 ps -ef|grep -i pmon
grid      3801     1  0 07:09 ?        00:00:13 apx_pmon_+APX2
grid      5973     1  0 07:45 ?        00:00:21 asm_pmon_+ASM2
oracle    7114     1  0 08:01 ?        00:00:24 ora_pmon_orcl_1
[oracle@host01 ~]$ ssh host03 ps -ef|grep -i pmon
grid      2981     1  0 13:49 ?        00:00:04 apx_pmon_+APX3
oracle    3034     1  0 13:49 ?        00:00:07 ora_pmon_orcl_2
grid      3682     1  0 13:51 ?        00:00:06 asm_pmon_+ASM3
[oracle@host01 ~]$
```

2. Display the syntax usage help for the `srvctl status home` command.

```
[oracle@host01 ~]$ srvctl status home -help

Displays the current state of of all resources for the Oracle
home.

Usage: srvctl status home -oraclehome <oracle_home> -statefile
<state_file> -node <node_name>
    -oraclehome <path>             Oracle home path
    -statefile <state_file>        Specify a file path for the
srvctl status home command to store the state of the resources
    -node <node_name>              Node name
    -help                          Print usage
[oracle@host01 ~]$
```

3. Use the srvctl status home command to check the state of all resources running from the `/u01/app/oracle/product/12.1.0/dbhome_1` home location. Create the required state file in the `/tmp` directory with the file name `host01_dbhome_state.dmp` for the first node only.

```
[oracle@host01 ~]$ srvctl status home -oraclehome
/u01/app/oracle/product/12.1.0/dbhome_1 -statefile
/tmp/host01_dbhome_state.dmp -node host01
Database orcl is running on node host01
[oracle@host01 ~]$
```

4. Display the syntax usage help for the srvctl stop home command.

```
[oracle@host01 ~]$ srvctl stop home -help

Stops all Oracle clusterware resources that run from the Oracle
home.

Usage: srvctl stop home -oraclehome <oracle_home> -statefile
<state_file> -node <node_name> [-stopoption <stop_options>] [-
force]
    -oraclehome <path>             Oracle home path
    -statefile <state_file>        Specify a file path for the
srvctl stop home command to store the state of the resources
    -node <node_name>              Node name
    -stopoption <stop_options>     Stop options for the
database. Examples of shutdown options are NORMAL,
TRANSACTIONAL, IMMEDIATE, or ABORT.
    -force                         Force stop
    -help                          Print usage
[oracle@host01 ~]$
```

5. Stop all resources executing from `/u01/app/oracle/product/12.1.0/dbhome_1`. Do not use the optional parameters identified by square brackets "`[]`" displayed in the syntax usage help.

```
[oracle@host01 ~]$ srvctl stop home -oraclehome
/u01/app/oracle/product/12.1.0/dbhome_1 -node host01 -statefile
/tmp/host01_dbhome_state1.dmp
[oracle@host01 ~]$
```

6. Check the status of the database instances on each node. .

```
[oracle@host01 ~]$ srvctl status database -d orcl
Instance orcl_1 is running on node host02
Instance orcl_2 is running on node host03
Instance orcl_3 is not running on node host01
[oracle@host01 ~]$
```

7. Start all resources for the `/u01/app/oracle/product/12.1.0/dbhome_1` home using the state file created by the stop command.

```
[oracle@host01 ~]$ srvctl start home -oraclehome
/u01/app/oracle/product/12.1.0/dbhome_1 -node host01 -statefile
/tmp/host01_dbhome_state1.dmp
[oracle@host01 ~]$
```

8. Check the status of the database instances on each node.

```
[oracle@host01 ~]$ srvctl status database -d orcl
Instance orcl_1 is running on node host02
Instance orcl_2 is running on node host03
Instance orcl_3 is running on node host01
[oracle@host01 ~]$
```

9. Close all terminal windows opened for this practice.

Practices for Lesson 5: Managing Backup and Recovery for RAC

Chapter 5

Practices for Lesson 5: Overview

Practices Overview

In this practice, you will configure ARCHIVELOG mode for your RAC database, configure instance-specific connect strings for RMAN, and configure persistent RMAN settings.

Practice 5-1: Configuring Archive Log Mode

Overview

In this practice, you adjust initialization parameters in the SPFILE, and stop and start the ASM instances on local and remote nodes.

1. Open a terminal session to `host01` as the `oracle` user and set up the environment variables using the `oraenv` script for the database instance. Determine the instance running on host01 (the local machine). Change the value of the `ORACLE_SID` variable to allow local system authenticated connections.

```
[vncuser@classroom_pc ~] ssh oracle@host01
Password: <oracle>
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base has been set to /u01/app/oracle

[oracle@host01 ~]$ srvctl status database -db orcl
Instance orcl_1 is running on node host02
Instance orcl_2 is running on node host03
Instance orcl_3 is running on node host01

[oracle@host01 ~]$ export ORACLE_SID=orcl_3
[oracle@host01 ~]$
```

2. Make a local connection using operating system authentication to the database instance, and then use the archive log list SQL command to determine whether the database is in archive log mode. Exit SQL*Plus when done.

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

SQL*Plus: Release 12.1.0.1.0 Production on Mon Jan 13 17:24:39
2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Enter password:
Last Successful login time: Mon Jan 13 2014 17:07:01 +00:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
```

```
SQL> archive log list
Database log mode              No Archive Mode
Automatic archival             Disabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     80
Current log sequence           81
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
[oracle@host01 ~]$
```

3. Stop the `orcl` database on each node of the cluster by using the `srvctl stop database` command.

```
[oracle@host01 ~]$ srvctl stop database -d orcl
[oracle@host01 ~]$
```

4. Verify that the `orcl` database is not running on any node of the cluster by using the srvctl status database command.

```
[oracle@host01 ~]$ srvctl status database -d orcl
Instance orcl_1 is not running on node host02
Instance orcl_2 is not running on node host03
Instance orcl_3 is not running on node host01
[oracle@host01 ~]$
```

5. Make a local connection using operating system authentication to the local database instance, and then start up the database on only the first node with the `mount` option.

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

SQL*Plus: Release 12.1.0.1.0 Production on Thu Sep 12 06:52:26
2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup mount
ORACLE instance started.
```

```
Total System Global Area 1018830848 bytes
Fixed Size                  2295992 bytes
Variable Size             427820872 bytes
Database Buffers          583008256 bytes
Redo Buffers                5705728 bytes
Database mounted.
SQL>
```

6. Issue the `alter database archivelog` SQL command to change the archive mode of the database, and then verify the results by using the archive log list SQL command.

```
SQL> alter database archivelog;

Database altered.

SQL> archive log list
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     80
Next log sequence to archive   81
Current log sequence           81
SQL>
```

7. Shut down the database instance with the `immediate` option and exit SQL*Plus. Use the `srvctl` utility to restart the database instances on all nodes of the cluster.

```
SQL> shutdown immediate
ORA-01109: database not open


Database dismounted.
ORACLE instance shut down.
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

[oracle@host01 ~]$ srvctl start database -d orcl
[oracle@host01 ~]$
```

8. Verify that the `orcl` database is running on all the three nodes of your cluster by using the `srvctl status database` command.

```
[oracle@host01 ~]$ srvctl status database -d orcl
Instance orcl_1 is running on node host03
Instance orcl_2 is running on node host02
Instance orcl_3 is running on node host01
[oracle@host01 ~]$
```

Practice 5-2: Configuring RMAN and Performing Parallel Backups

Overview

In this practice, you will designate the first and second instances (`orcl_1` and `orcl_2`) of your policy-managed database responsible for performing parallel backups of the database. The database will be backed up to the +FRA ASM disk group by default.

1. Using the recovery manager utility (RMAN), connect to the `orcl` database as the target database.

```
[oracle@host01 ~]$ rman target /

Recovery Manager: Release 12.1.0.1.0 - Production on Thu Sep 12
09:03:24 2013

Copyright (c) 1982, 2013, Oracle and/or its affiliates.  All
rights reserved.

connected to target database: ORCL (DBID=1352492209)

RMAN>
```

2. Display all of the current RMAN settings.

```
RMAN> show all;

using target database control file instead of recovery catalog
RMAN configuration parameters for database with db_unique_name
ORCL are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # 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/12.1.0/dbhome_1/dbs/snapcf_orcl_3.f'; #
default

RMAN>
```

3. Configure RMAN to automatically back up the control file and server parameter file each time any backup operation is performed.

```
RMAN> configure controlfile autobackup on;

new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored

RMAN>
```

4. Configure channels to use automatic load balancing. Set parallelism to 2.

```
RMAN> CONFIGURE DEVICE TYPE disk PARALLELISM 2;

new RMAN configuration parameters:
CONFIGURE DEVICE TYPE DISK PARALLELISM 2 BACKUP TYPE TO
BACKUPSET;
new RMAN configuration parameters are successfully stored

RMAN>
```

5. Open a second terminal session as the `oracle` user and set up the environment variables for the `orcl` database. Invoke SQL*plus as the system user, and run the `/stage/RAC/labs/less_05/monitor_rman.sql` script. Do not exit the first session with the RMAN prompt or this second session with the SQL prompt.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$ export ORACLE_SID=orcl_3
[oracle@host01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.1.0 Production on Thu Sep 12 09:16:51
2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.
```

```
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

SQL> @/stage/RAC/labs/less_05/monitor_rman.sql

no rows selected

SQL>
```

6. In the first session with the RMAN prompt, perform a full database backup with archive logs. The backup should happen only on the designated nodes (your first and second nodes) as the backup nodes. **Do not wait for this step to finish before proceeding to the next step.**

```
RMAN> backup database plus archivelog;

Starting backup at 12-SEP-13
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=39 instance=orcl_3 device type=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: SID=97 instance=orcl_3 device type=DISK
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=3 sequence=81 RECID=1 STAMP=825922793
input archived log thread=1 sequence=23 RECID=2 STAMP=825922794
channel ORA_DISK_1: starting piece 1 at 12-SEP-13
channel ORA_DISK_2: starting archived log backup set
channel ORA_DISK_2: specifying archived log(s) in backup set
input archived log thread=2 sequence=25 RECID=3 STAMP=825922794
input archived log thread=3 sequence=82 RECID=4 STAMP=825931183
input archived log thread=1 sequence=24 RECID=6 STAMP=825931055
input archived log thread=2 sequence=26 RECID=5 STAMP=825931134
input archived log thread=3 sequence=83 RECID=9 STAMP=825931464
input archived log thread=1 sequence=25 RECID=8 STAMP=825931462
input archived log thread=2 sequence=27 RECID=7 STAMP=825931407
input archived log thread=2 sequence=28 RECID=11 STAMP=825931981
input archived log thread=1 sequence=26 RECID=12 STAMP=825931901
channel ORA_DISK_2: starting piece 1 at 12-SEP-13
```

```
channel ORA_DISK_1: finished piece 1 at 12-SEP-13
piece
handle=+FRA/ORCL/BACKUPSET/2013_09_12/annnf0_tag20130912t094420_
0.285.825932663 tag=TAG20130912T094420 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=3 sequence=84 RECID=10 STAMP=825932033
input archived log thread=3 sequence=85 RECID=13 STAMP=825932243
input archived log thread=2 sequence=29 RECID=14 STAMP=825932191
input archived log thread=1 sequence=27 RECID=15 STAMP=825932111
input archived log thread=3 sequence=86 RECID=17 STAMP=825932655
input archived log thread=2 sequence=30 RECID=18 STAMP=825932602
input archived log thread=1 sequence=28 RECID=16 STAMP=825932520
channel ORA_DISK_1: starting piece 1 at 12-SEP-13
channel ORA_DISK_2: finished piece 1 at 12-SEP-13
piece
handle=+FRA/ORCL/BACKUPSET/2013_09_12/annnf0_tag20130912t094420_
0.286.825932667 tag=TAG20130912T094420 comment=NONE
channel ORA_DISK_2: backup set complete, elapsed time: 00:00:05
channel ORA_DISK_1: finished piece 1 at 12-SEP-13
piece
handle=+FRA/ORCL/BACKUPSET/2013_09_12/annnf0_tag20130912t094420_
0.287.825932673 tag=TAG20130912T094420 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 12-SEP-13

Starting backup at 12-SEP-13
using channel ORA_DISK_1
using channel ORA_DISK_2
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00003
name=+DATA/ORCL/DATAFILE/sysaux.282.824711255
input datafile file number=00006
name=+DATA/ORCL/DATAFILE/users.280.824711417
input datafile file number=00002
name=+DATA/ORCL/DATAFILE/example.276.824711595
channel ORA_DISK_1: starting piece 1 at 12-SEP-13
channel ORA_DISK_2: starting full datafile backup set
channel ORA_DISK_2: specifying datafile(s) in backup set
input datafile file number=00001
name=+DATA/ORCL/DATAFILE/system.281.824711341
```

```
input datafile file number=00004
name=+DATA/ORCL/DATAFILE/undotbs1.275.824711417
input datafile file number=00007
name=+DATA/ORCL/DATAFILE/undotbs3.268.824712405
input datafile file number=00005
name=+DATA/ORCL/DATAFILE/undotbs2.284.824712401
channel ORA_DISK_2: starting piece 1 at 12-SEP-13
channel ORA_DISK_2: finished piece 1 at 12-SEP-13
piece
handle=+FRA/ORCL/BACKUPSET/2013_09_12/nnndf0_tag20130912t094433_
0.289.825932687 tag=TAG20130912T094433 comment=NONE
channel ORA_DISK_2: backup set complete, elapsed time: 00:03:18
channel ORA_DISK_1: finished piece 1 at 12-SEP-13
piece
handle=+FRA/ORCL/BACKUPSET/2013_09_12/nnndf0_tag20130912t094433_
0.288.825932675 tag=TAG20130912T094433 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:03:39
Finished backup at 12-SEP-13

Starting backup at 12-SEP-13
current log archived
using channel ORA_DISK_1
using channel ORA_DISK_2
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=29 RECID=21 STAMP=825932765
input archived log thread=3 sequence=87 RECID=19 STAMP=825932898
channel ORA_DISK_1: starting piece 1 at 12-SEP-13
channel ORA_DISK_2: starting archived log backup set
channel ORA_DISK_2: specifying archived log(s) in backup set
input archived log thread=2 sequence=31 RECID=20 STAMP=825932845
channel ORA_DISK_2: starting piece 1 at 12-SEP-13
channel ORA_DISK_1: finished piece 1 at 12-SEP-13
piece
handle=+FRA/ORCL/BACKUPSET/2013_09_12/annnf0_tag20130912t094821_
0.269.825932903 tag=TAG20130912T094821 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:02
channel ORA_DISK_2: finished piece 1 at 12-SEP-13
piece
handle=+FRA/ORCL/BACKUPSET/2013_09_12/annnf0_tag20130912t094821_
0.266.825932903 tag=TAG20130912T094821 comment=NONE
channel ORA_DISK_2: backup set complete, elapsed time: 00:00:01
Finished backup at 12-SEP-13
```

```
Starting Control File and SPFILE Autobackup at 12-SEP-13
piece
handle=+FRA/ORCL/AUTOBACKUP/2013_09_12/s_825932904.267.825932907
comment=NONE
Finished Control File and SPFILE Autobackup at 12-SEP-13

RMAN>
```

7. While the backup is in progress, rerun the query on the second terminal window to monitor the RMAN backup session progress within the cluster. The backup should be done in parallel, with work distributed to both the backup nodes of the cluster. Enter the slash (/) symbol and press the Enter key to rerun the query. It may be necessary to do this multiple times until the output appears. When the backup finishes, exit SQL*Plus.

```
SQL> /

INST_ID    SID    SERIAL#  CONTEXT   SOFAR  TOTALWORK %_COMPLETE
------- ------ ---------- -------- ------- ---------- ----------
      3     39       1051        1   26747     238400      11.22
      3     97        605        1       0     172160          0

SQL> /

INST_ID    SID    SERIAL#  CONTEXT   SOFAR  TOTALWORK %_COMPLETE
------- ------ ----------  ------- ------- ---------- ----------
      3     39       1051        1   66427     238400      27.86
      3     97        605        1   79337     172160      46.08

SQL> /

INST_ID     SID   SERIAL#  CONTEXT   SOFAR  TOTALWORK %_COMPLETE
---------- ---- --------- -------- ------- ---------- ----------
         3   39      1051        1  129980     238400      54.52
         3   97       605        1  127593     172160      74.11

SQL> exit

[oracle@host01 ~]$
```

8. Disable ARCHIVELOG mode for your RAC database. Shut down the database using `srvctl`.

```
[oracle@host01 ~]$ srvctl stop database -d orcl
[oracle@host01 ~]$
```

9. Make a local connection using operating system authentication to the local database instance, and then start up the database on only the first node with the `mount` option. Disable archivelog mode with the `alter database noarchivelog` statement. Confirm this operation with the archive log list statement. Shut down the database and exit SQL*Plus when finished.

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

SQL*Plus: Release 12.1.0.1.0 Production on Thu Sep 12 12:30:00
2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup mount
ORACLE instance started.

Total System Global Area 1018830848 bytes
Fixed Size                  2295992 bytes
Variable Size             427820872 bytes
Database Buffers          583008256 bytes
Redo Buffers                5705728 bytes
Database mounted.
SQL> alter database noarchivelog;

Database altered.

SQL> archive log list
Database log mode              No Archive Mode
Automatic archival             Disabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     87
Current log sequence           88

SQL> shutdown immediate;
ORA-01109: database not open
```

```
Database dismounted.
ORACLE instance shut down.
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
[oracle@host01 ~]$
```

10. Use `srvctl` to re-start your database. Ensure that all instances are up, and then exit all terminal windows.

```
[oracle@host01 ~]$ srvctl start database -d orcl

[oracle@host01 ~]$ srvctl status database -d orcl
Instance orcl_1 is running on node host03
Instance orcl_2 is running on node host02
Instance orcl_3 is running on node host01
[oracle@host01 ~]$
```

11. Close all terminal windows opened for this practice.

Practices for Lesson 6: Global Resource Management Concepts

Chapter 6

Practices for Lesson 6: Overview

Practices Overview

In this practice, you will install Enterprise Manager agents on host01 and host02 in preparation for Lesson 7 practices. You will also perform target discovery.

Practice 6-1: Pre-Practice Tasks

Overview

Before starting the lesson 7 practices, you must configure Enterprise Manager Cloud Control. You must start agents on your monitored host, start the server hosting Enterprise Manager and complete target discovery.

1. Open a terminal session on your desktop. Become the `root` user and shut down `host03`. Do not proceed to the next step until `host03` has been shut down. When `host03` has been shut down, start host `em12`.

```
[vncuser@classroom_pc ~]$ su -
Password: <oracle>
[root@EDRSR46P1 ~]# xm shutdown host03

[root@EDRSR46P1 ~]# xm list
Name                        ID   Mem VCPUs      State   Time(s)
Domain-0                     0  1044     2     r-----   9253.8
host01                       8  4200     1     -b----  22219.2
host02                       9  3200     2     -b----  25065.7
host03                      14  3200     2     -b----  11860.2

[root@EDRSR46P1 ~]# xm list
Name                        ID   Mem VCPUs      State   Time(s)
Domain-0                     0  1044     2     r-----   9253.8
host01                       8  4200     1     -b----  22219.2
host02                       9  3200     2     -b----  25065.7

[root@EDRSR46P1 ~]# xm create em12
Using config file "/etc/xen/em12".
Started domain em12 (id=6)

[root@EDRSR46P1 ~]# xm list
Name                        ID   Mem VCPUs      State   Time(s)
Domain-0                     0  1044     2     r-----   9253.8
host01                       8  4200     1     -b----  22219.2
host02                       9  3200     2     -b----  25065.7
em12                        10  6500     2     -b----      5.1
```

2. Wait a few minutes and then open a terminal to the `em12` host as the `oracle` user. Be sure to use the `-X` option. Check the status of the OMS. Proceed to the next step when both the WebTier and Oracle Management Server are up. This could take several minutes.

```
[root@classroom_pc ~]# ssh -X oracle@em12
oracle@em12's password: <oracle>
```

```
Last login: Tue Apr  9 12:58:10 2013 from 192.0.2.1

[oracle@em12 ~]$ /u01/em121020/oms/bin/emctl status oms
Oracle Enterprise Manager Cloud Control 12c Release 2
Copyright (c) 1996, 2012 Oracle Corporation.  All rights
reserved.
WebTier is Up
Oracle Management Server is Up
[oracle@em12 ~]$
```

3. Open a terminal session to `host01` as the `oracle` user. Connect to the database and unlock the `dbsnmp` user.

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

[oracle@host01 ~]$ sqlplus sys/oracle_4U@orcl as sysdba

SQL*Plus: Release 12.1.0.1.0 Production on Wed Sep 25 07:12:53
2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.



Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

SQL> alter user dbsnmp identified by oracle_4U account unlock;

User altered.

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
[oracle@host01 ~]$
```

4. Switch user to the `grid` account. Set the environment and start a SQL*Plus session as `SYSASM`. Make sure the `ASMSNMP` user exists and the account is unlocked. If it doesn't,

create it with a password of `oracle_4U` and `SYSDBA` privileges. Exit when finished, and go back to the Oracle account.

```
[oracle@host01 ~]$ su - grid
Password:
[grid@host01 ~]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base has been set to /u01/app/grid

[grid@host01 ~]$ sqlplus / as sysasm

SQL*Plus: Release 12.1.0.1.0 Production on Mon Jan 27 14:20:09
2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Real Application Clusters and Automatic Storage
Management options

SQL> select * from v$pwfile_users;

USERNAME         SYSDB SYSOP SYSAS SYSBA SYSDG SYSKM     CON_ID
---------------- ----- ----- ----- ----- ----- ----- ----------
SYS              TRUE  TRUE  FALSE FALSE FALSE FALSE          0
CRSUSER__ASM_001 TRUE  FALSE TRUE  FALSE FALSE FALSE          0
ASMSNMP          TRUE  FALSE FALSE FALSE FALSE FALSE          0

******* If the ASMSNMP user does not exist, create it! *********

SQL> create user asmsnmp identified by oracle_4U;
SQL> grant sysdba to asmsnmp;
Grant Succeeded

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Real Application Clusters and Automatic Storage
Management options

[grid@host01 ~]$ exit
logout
```

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

5. From the oracle window on `em12`, start a browser and enter the following address:
   https://em12:7803/em

```
[oracle@em12 ~]$ firefox&
[1] 32419
[oracle@em12 ~]$
```

6. Add a security exception, if prompted and log in to Enterprise Manager as `sysman`/`oracle_4U`.

7. Select Setup > Add Target > Add Targets Manually.

8. On the Select Targets Manually page, select Add Host Targets then click Add Hosts.

9. Click +Add and type in `host01.example.com`. Click +Add and type in `host02.example.com`. Select Same for All Hosts from platform pull-down menu, select Linux x86-64 from the Platform pull down menu and click Next

10. On the Add Host Targets: Installation Details page, enter the following:
    Installation Base Directory: **/u01/app/oracle/agent**
    Instance Directory: /**u01/app/oracle/agent/agent_inst**
    Named Credential: Select **NC_HOST_2013-04-12-131546**
    Privileged Delegation Setting: Make this field blank (run root scripts manually)
    Port: **3872**
    Click Next.

11. Click "Deploy Agent" button. It will take several minutes for the agent software to be pushed to the nodes.

12. Keep track of the progress in the Agent Deployment Details chart at the bottom the page. When prompted from the Recommendation column, run the `root` scripts. Go to the `oracle` terminal window opened on `host01` and become the `root` user.

```
[oracle@host01 oracle]$ su -
Password: <oracle>
[root@host01 ~]# /u01/app/oracle/agent/core/12.1.0.2.0/root.sh
Finished product-specific root actions.
/etc exist
Finished product-specific root actions.

[root@host01 ~]# ssh host02
/u01/app/oracle/agent/core/12.1.0.2.0/root.sh
Finished product-specific root actions.
/etc exist
Finished product-specific root actions.
[root@host01 ~]# exit

[oracle@host01 ~]$
```

When the scripts have been run on both nodes, click Done on the Add Host Status page.

13. Navigate to Targets > Hosts. Make sure `host01` and `host02` are listed with a status of Up.

14. Next, navigate to Setup > Add Target > Add Targets Manually. Select "Add Non-Host Targets Using Guided Process (Also Adds Related Targets)". Select Oracle Cluster and High Availability Service from the Target Types pull-down menu and click "Add Using Guided Discovery".

15. On the Add Cluster Target: Specify Host page enter **`host01.example.com`**. Click Continue.

16. On the Add Target: Cluster page. Make sure that the fields contain the following information:

    Cluster Name: **`cluster01`**

    Oracle Home: **`/u01/app/12.1.0/grid`**

    SCAN Name: **`cluster01-scan.cluster01.example.com`**

    SCAN Port: **`1521`**

    ONS Port: **`2016`**

    Selected Hosts**:** **`host01.example.com`** and **`host02.example.com`**.

    Click Add. When finished, click OK.

17. Select Targets > All Targets. Make sure that `cluster01`, `has_host01.example.com`, and `has_host02.example.com` appear in the Targets column with a status of Up.

18. Go to Setup > Add Target > Add Targets Manually. Select Add Non-Host Targets Using Guided Process (Also Adds Related Targets). Select Oracle Database, Listener, and Automatic Storage Management, and click Add Using Guided Discovery.

19. On the Add Database Instance Target: Specify Host page enter host01.example.com. Click Continue.

20. On the Add Database: Specify Source page, select "on all hosts in the cluster," and click Continue.

21. On the Discovered Targets on Cluster: cluster01 page, find the `orcl` database and enter `oracle_4U` in the Monitor Password field. Enter `Oracle_4U` in the Cluster ASM Monitor Password field. Click the Test Connection button. If both connections are successful, click Finish, then click Save. Click OK on the Target Configuration results.

22. Select Targets > All Targets. Make sure that the `orcl` cluster database and cluster ASM targets appear in the list with their associated instances with a status of Up.

Practices for Lesson 7: RAC Database Monitoring and Tuning

Chapter 7

Practices for Lesson 7: Overview

Practices Overview

This practice is designed to show you how to discover performance problems in your RAC environment. In this practice, you identify performance issues by using Enterprise Manager, and fix issues in three different steps. At each step, you will generate the same workload to make sure that you are making progress in your resolution.

Practice 7-1: ADDM and RAC Part I

Overview

The goal of this practice is to show you how to manually discover performance issues by using the Enterprise Manager performance pages as well as ADDM. This first part generates a workload that uses a bad RAC application design.

Note that all the necessary scripts for this practice are located in the `/stage/RAC/labs/less_07` directory.

1. Connect to your first node as the `oracle` user and set up your environment variables using the `oraenv` script. Determine the oracle instance running on `host01`. You will need this information throughout these practices.

```
[vncuser@classroom_pc ~]$ ssh oracle@host01
oracle@host01's password:
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base has been set to /u01/app/oracle

[oracle@host01 ~]$ ps -ef|grep ora_smon
oracle   18660  3459  0 17:25 pts/3    00:00:00 grep ora_smon
oracle   31630     1  0 Jan13 ?        00:01:06 ora_smon_orcl_3
[oracle@host01 ~]$
```

2. Execute the `setupseq1.sh` script from the `/stage/RAC/labs/less_07` directory to set up the necessary configuration for this practice.

```
[oracle@host01 ~]$ cd /stage/RAC/labs/less_07

[oracle@host01 less_07]$ ./setupseq1.sh

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.

 drop user jmw cascade
           *
ERROR at line 1:
ORA-01918: user 'JMW' does not exist


 drop tablespace seq including contents and datafiles
*
ERROR at line 1:
```

```
ORA-00959: tablespace 'SEQ' does not exist



Tablespace created.


User created.


Grant succeeded.

 drop sequence s
               *
ERROR at line 1:
ORA-02289: sequence does not exist


 drop table s purge
            *
ERROR at line 1:
ORA-00942: table or view does not exist


 drop table t purge
            *
ERROR at line 1:
ORA-00942: table or view does not exist



Table created.


Table created.


Index created.


1 row created.
```

```
Commit complete.


PL/SQL procedure successfully completed.

[oracle@host01 less_07]$
```

3. Return to Enterprise Manager and navigate to the Performance page of your Cluster Database. Click Targets > All Targets > orcl.
   1) Click the Performance tab from the Cluster Database Home page and click Performance Home from the pull down menu. If prompted to log in to the database, enter `sys/oracle_4U` as `sysdba`.
   2) On the Cluster Database Performance page, make sure that Real Time:15 Seconds Refresh is selected from the View Data pull-down list.

4. Use PL/SQL to create a new AWR snapshot.

```
[oracle@host01 less_07]$ ./create_snapshot.sh

PL/SQL procedure successfully completed.

[oracle@host01 less_07]$
```

5. Open a second terminal to `host01` as the `oracle` user. Change directory to `/stage/RAC/labs/less_07`. Execute the `lockinfo.sh` script. This script allows you to view global lock contention issues. Your output should show no transactions for JMW.

```
[vncuser@classroom_pc ~]$ ssh oracle@host01
oracle@host01's password:

[oracle@host01 ~]$ cd /stage/RAC/labs/less_07

[oracle@host01 less_07]$ ./lockinfo.sh

SQL*Plus: Release 12.1.0.1.0 Production on Mon Sep 16 14:46:21
2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
```

```
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

SQL>
Sess          Op Sys              OBJ NAME or
 ID  USERNAME User ID TERMINAL TRANS_ID     TY Lock Mode   Req Mode
---- -------- ------- -------- ------------ -- ----------- -----------
  46 SYS      oracle     unknown  ORA$BASE     AE Share
  48 SYS      oracle              ORA$BASE     AE Share

SQL> Disconnected from Oracle Database 12c Enterprise Edition
Release 12.1.0.1.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
[oracle@host01 less_07]$
```

6. From the first `oracle` terminal, execute the `startseq1.sh` script to generate a workload on all instances of your cluster. **Do not wait; proceed with the next step**.

```
[oracle@host01 less_07]$ ./startseq1.sh

INSTANCE_NAME
----------------
orcl_3

INSTANCE_NAME
----------------
orcl_2

[oracle@host01 less_07]$
PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


[oracle@host01 less_07]$
```

7. From the second terminal, execute the `lockinfo.sh` script again to view information regarding possible lock contention. You can also select Blocking Sessions from the Cluster Database Performance pull down menu. You may have to refresh several times to see a lock contention for transactions belonging to JMW.

```
[oracle@host01 less_07]$ ./lockinfo.sh

SQL*Plus: Release 12.1.0.1.0 Production on Mon Sep 16 15:09:45
2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.



Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

SQL>
Sess          Op Sys           OBJ NAME or
 ID  USERNAME User ID TERMINAL TRANS_ID     TY Lock Mode   Req Mode
---- -------- ------- -------- ------------ -- ----------- -----------
  35 JMW      oracle  pts/1    ORA$BASE     AE Share
  37 DBSNMP   oracle  unknown  ORA$BASE     AE Share
  41 DBSNMP   oracle  unknown  ORA$BASE     AE Share
  43 JMW      oracle  pts/1    Trans-2      PS Share
  44 SYS      oracle  unknown  ORA$BASE     AE Share
  47 JMW      oracle           ORA$BASE     AE Share
  47 JMW      oracle           T            TM Row Excl
  47 JMW      oracle           S            TM Exclusive
  47 JMW      oracle           Trans-262171 TX Exclusive
  60 SYS      oracle  unknown  ORA$BASE     AE Share
  73 DBSNMP   oracle  unknown  ORA$BASE     AE Share
  80 DBSNMP   oracle  unknown  ORA$BASE     AE Share
  86 DBSNMP   oracle  unknown  ORA$BASE     AE Share
  88 JMW      oracle           ORA$BASE     AE Share
  88 JMW      oracle           S            TM --Waiting-- Exclusive
  90 SYS      oracle  unknown  ORA$BASE     AE Share
  91 JMW      oracle  pts/1    ORA$BASE     AE Share
  91 JMW      oracle  pts/1    Trans-3      PS Share
  91 JMW      oracle  pts/1    Trans-2      PS Share
  93 DBSNMP   oracle  unknown  ORA$BASE     AE Share
  94 DBSNMP   oracle  unknown  ORA$BASE     AE Share
  96 DBSNMP   oracle  unknown  ORA$BASE     AE Share
  98 DBSNMP   oracle  unknown  ORA$BASE     AE Share
  99 DBSNMP   oracle  unknown  ORA$BASE     AE Share
 101 DBSNMP   oracle  unknown  ORA$BASE     AE Share
 102 JMW      oracle  pts/1    Trans-3      PS Share
```

```
 104 DBSNMP   oracle  unknown  ORA$BASE     AE Share

27 rows selected.


SQL> Disconnected from Oracle Database 12c Enterprise Edition
Release 12.1.0.1.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
[oracle@host01 less_07]$
```

8. While the scripts are still executing, look at the Average Active Sessions graphic. Then, drill down to the Cluster wait class for the first node. What are your conclusions?
   1) Return to the Cluster Database Performance page.
   2) From there you can now see the Average Active Sessions graph. Make sure that the View Data field is set to Real Time:15 Seconds Refresh. After a few seconds, the graphic should clearly show that the Cluster and Application wait classes are causing most waits.
   3) In the Average Active Sessions graph, click the Cluster link on the right. This takes you to the Active Sessions By Instance: Cluster page.
   4) On the Active Sessions By Instance: Cluster page, you will see that the number of active sessions is almost the same on all nodes. Click the instance running on host01 (`orcl_3` in this example). This takes you to the Active Sessions Waiting: Cluster page for the corresponding instance.
   5) On the Active Sessions Waiting: Cluster page, you can see the most important wait events causing most of the waits in the Cluster wait class on the first instance (you may have to refresh the page a time or two). In the Top SQL: Cluster section, click the SQL identifier that uses most of the resources. This takes you to the SQL Details page for the corresponding statement. You will see that the script running on the first instance is executing a `SELECT/UPDATE` statement on table S that causes most of the Cluster waits.

   By using the drill-down method of Enterprise Manager, you can quickly identify the top waiting SQL statements and the top waiting sessions on both instances. Here it appears that a SELECT or UPDATE statement on table S is causing most of the waits for the Cluster wait class.

9. Using Enterprise Manager, look at the Cluster Cache Coherency page. What are your conclusions?
   1) On the Cluster Database Home page, select the Cluster Cache Coherency link click from the Performance pull-down menu.
   2) The Cluster Cache Coherency page clearly shows that there are lots of blocks transferred per second on the system. This represents more than 17% of the total logical reads. This is reflected in both the Global Cache Block Transfer Rate and the Global Cache Block Transfers and Physical Reads (vs. Logical Reads) graphics.
   3) On the Cluster Cache Coherency page, you can also click Interconnects in the Additional Links section of the page to get more information about your private interconnect.

10. While the scripts are still executing, look at the Average Active Sessions graph on the Database Performance page. Follow the instructions below. What are your conclusions?
    1) Drill down to the Application wait class and then drill down on the `host01` instance.
    2) By using the drill-down method of Enterprise Manager, you can quickly identify the top waiting SQL statements and the top waiting sessions on both instances. Here it appears that a LOCK statement on table S is causing most of the waits for the Application wait class.
    3) Go back to the Cluster Database Home page and click the Performance tab and select Performance Home.
    4) Make sure that the View Data field is set to Real Time: 15 Seconds Refresh. After a few seconds, the graphic should clearly show that the Cluster and Application wait classes are causing most waits. You will also notice that the transaction rate is about 100 per second as shown on the Throughput tab.
    5) In the Average Active Sessions graph, click the Application link on the right. This takes you to the Active Sessions By Instance: Application page.
    6) On the Active Sessions By Instance: Application page, you must see that the number of active sessions is almost the same on all nodes. Click the link for the `host01` instance, orcl_3 in this case on the Summary Chart graph. This takes you to the Active Sessions Waiting: Application page of the host01 instance.
    7) On the Active Sessions Waiting: Application page, you can see the most important wait events causing most of the waits in the Application wait class on the host01 instance. In the Top SQL: Application section, click the SQL identifier that uses most of the resources. This takes you to the SQL Details page for the corresponding statement. You must see that the script running on the first instance is executing a LOCK statement on table S that causes most of the Application waits.
    8) After a while, you can see that all scripts are executed by looking at the Average Active Sessions graph as well as the Database Throughput graphics again. You should see the number of transactions per second going down.

11. After the workload finishes, use PL/SQL to create a new AWR snapshot.

```
[oracle@host01 less_07]$ ./create_snapshot.sh

PL/SQL procedure successfully completed.

[oracle@host01 less_07]$
```

12. Using Enterprise Manager, review the latest ADDM run. What are your conclusions?
    1) On the Cluster Database Home page, select the Advisors Home sub-menu from the Performance pull down menu.
    2) In the Results table, select the latest ADDM run corresponding to Instance All. Then click View Result. This takes you to the Automatic Database Diagnostic Monitor (ADDM) page.
    3) On the Automatic Database Diagnostic Monitor (ADDM) page, the ADDM Performance Analysis table shows you the consolidation of ADDM reports from all instances running in your cluster. This is your first entry point before drilling down to specific instances.

From there, investigate the Top SQL Statements, Table Locks, and Global Cache Messaging findings.

4) Click the Top SQL Statements finding, which affects all instances, revealing LOCK TABLE S and UPDATE S commands as a possible problem to investigate. Click the Back button to return to the ADDM report.
5) Click the Table Locks finding, which affects all instances, revealing that you should investigate your application logic regarding the JMW.S object.
6) Click the Back button to return to the ADDM report. Click the Global Cache Messaging finding revealing either the UPDATE S or SELECT command as responsible for approximately 30% of Cluster waits during the analysis period.
7) You now have the possibility to drill down to each instance using the links located in the Affected Instances table. Click the link corresponding to the most affected instance (although all should be equally affected).
8) On the corresponding ADDM Database Diagnostic Monitor (ADDM) instance page, you should retrieve similar top findings you previously saw at the cluster level.

Practice 7-2: ADDM and RAC Part II

Overview

The goal of this practice is to show you how to manually discover performance issues by using the Enterprise Manager performance pages as well as ADDM. In this second part of the practice, you are going to correct the previously found issue by creating a sequence number instead of by using a table.

Note that all the necessary scripts for this practice are located in the `/stage/RAC/labs/less_07` directory.

1. Execute the `setupseq2.sh` script to create the necessary objects used for the rest of this practice.

```
[oracle@host01 less_07]$ ./setupseq2.sh

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


User dropped.


Tablespace dropped.


Tablespace created.


User created.


Grant succeeded.

 drop table s purge
            *
ERROR at line 1:
ORA-00942: table or view does not exist


 drop sequence s
                *
ERROR at line 1:
```

```
ORA-02289: sequence does not exist


 drop table t purge
            *
ERROR at line 1:
ORA-00942: table or view does not exist



Table created.


Index created.


Sequence created.


PL/SQL procedure successfully completed.

[oracle@host01 less_07]$
```

2. Using Enterprise Manager, and connected as the SYSMAN user, navigate to the Performance page of your Cluster Database.
   1) Click the Performance tab from the Cluster Database Home page and select Performance Home.
   2) On the Cluster Database Performance page, make sure Real Time: 15 Seconds Refresh is selected from the View Data drop-down list.

3. Use PL/SQL to create a new AWR snapshot.

```
[oracle@host01 less_07]$ ./create_snapshot.sh

PL/SQL procedure successfully completed.

[oracle@host01 less_07]$
```

4. Execute the `startseq2.sh` script to generate a workload on all instances of your cluster. **Do not wait; proceed with the next step**.

```
[oracle@host01 less_07]$ ./startseq2.sh

… Do not wait after this point and go to the next step.

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed
[oracle@host01 less_07]$
```

5. While the scripts are still executing, look at the Average Active Sessions graphic. Then drill down to the Cluster wait class for the first node. What are your conclusions?
   1) Return to the Cluster Database Performance Home page.
   2) From there you can now see the Average Active Sessions graph. Make sure that the View Data field is set to Real Time:15 Seconds Refresh. After a few seconds, the graphic will clearly show that the Cluster and Application wait classes are causing most waits. Using the Throughput tabbed page graph underneath the Average Active Sessions graph, you should also notice that the transaction rate is about 200 or more per second (a better rate than in the previous practice).
   3) In the Average Active Sessions graph, click the Cluster link on the right. This takes you to the Active Sessions By Instance: Cluster page.
   4) On the Active Sessions By Instance: Cluster page, you should see that the number of active sessions is almost the same on all nodes. Click the first instance's link, orcl_3 in this case (on `host01`). This takes you to the Active Sessions Waiting: Cluster page for the corresponding instance.
   5) On the Active Sessions Waiting: Cluster page, you can see the most important wait events causing most of the waits in the Cluster wait class on the first instance. In the Top SQL: Cluster section, click the SQL identifier that uses most of the resources. This takes you to the SQL Details page for the corresponding statement. You will see that the script running on the first instance is executing an INSERT statement on table T that causes most of the Cluster waits.
   6) After a while you can see that all transactions are executed by looking at the Average Active Sessions graphic again. Check the oracle terminal window where the workload was started. You should see two "PL/SQL procedure successfully completed." messages, indicating the workload has finished. The Database Throughput graphic tells you that this time, the number of transactions per second was a bit higher than in the previous practice for the same workload. Using the sequence number was a bit better in this case.

6. After the workload finishes, use PL/SQL to create a new AWR snapshot.

```
[oracle@host01 less_07]$ ./create_snapshot.sh

PL/SQL procedure successfully completed.

[oracle@host01 less_07]$
```

7. Using Enterprise Manager, review the latest ADDM run. What are your conclusions?
   1) On the Cluster Database Performance pull down menu, click the Advisors Home sub-menu.
   2) In the Results table, select the latest ADDM run corresponding to Instance All. Then click View Result. This takes you to the Automatic Database Diagnostic Monitor (ADDM) page.
   3) On the Automatic Database Diagnostic Monitor (ADDM) page, the ADDM Performance Analysis table shows you the consolidation of ADDM reports from all instances running in your cluster. This is your first entry point before drilling down to specific instances. From there, investigate the Top SQL Statements, Sequence Usage, and Unusual "Concurrency" Wait Event findings.
   4) The Top SQL Statements should reveal an INSERT INTO T command using sequence S as a possible problem to investigate.
   5) The Sequence Usage finding reveals that you should use larger cache size for your hot sequences.
   6) The Unusual "Concurrency" Wait Event finding asks you to investigate the cause for high "row cache lock" waits. Refer to the Oracle Database Reference for the description of this wait event.

Practice 7-3: ADDM and RAC Part III

Overview

The goal of this practice is to show you how to manually discover performance issues by using the Enterprise Manager performance pages as well as ADDM. This last part generates the same workload as in the previous practice, but uses more cache entries for sequence number S.

Note that all the necessary scripts for this practice are located in the `/stage/RAC/labs/less_07` directory.

1. Execute the `setupseq3.sh` script to create the necessary objects used for the rest of this practice.

```
[oracle@host01 less_07]$ ./setupseq3.sh

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


User dropped.


Tablespace dropped.


Tablespace created.


User created.


Grant succeeded.

 drop table s purge
            *
ERROR at line 1:
ORA-00942: table or view does not exist


 drop sequence s
                *
ERROR at line 1:
```

```
ORA-02289: sequence does not exist


 drop table t purge
            *
ERROR at line 1:
ORA-00942: table or view does not exist



Table created.


Index created.


Sequence created.


PL/SQL procedure successfully completed.

[oracle@host01 less_07]$
```

2. Using Enterprise Manager, navigate to the Performance page of your Cluster Database.
   1) Click the Performance tab from the Cluster Database Home page and select Performance Home.
   2) On the Cluster Database Performance page, make sure that Real Time: 15 Seconds Refresh is selected from the View Data drop-down list.

3. Use PL/SQL to create a new AWR snapshot.

```
[oracle@host01 less_07]$ ./create_snapshot.sh

PL/SQL procedure successfully completed.

[oracle@host01 less_07]$
```

4. Execute the `startseq2.sh` script to generate the same workload on both instances of your cluster as for the previous practice. Do not wait, and proceed with the next step.

```
[oracle@host01 less_07]$ ./startseq2.sh


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.

[oracle@host01 less_07]$
```

5. Until the scripts are executed, look at the Average Active Sessions graphic. What are your conclusions?
   1) On the Performance page, make sure that the View Data field is set to Real Time:15 Seconds Refresh. After all the scripts have finished their execution, the Average Active Sessions graph will clearly show that there are no significant waits on your cluster. You must also notice that the transaction rate is now around 1600 per second.

      This time, looking at the Average Active Sessions graphic, it is clear that there are no significant waits. The sequence has a big enough cache value to avoid the most significant waits.

6. After the workload finishes, use PL/SQL to create a new AWR snapshot.

```
[oracle@host01 less_07]$ ./create_snapshot.sh

PL/SQL procedure successfully completed.

[oracle@host01 less_07]$
```

7. Using Enterprise Manager, review the latest ADDM run. What are your conclusions?
   1) On the Cluster Database Home page, select Advisors Home from the Performance pull down menu.
   2) In the Results table, select the latest ADDM run corresponding to Instance All. Then click View Result. This takes you to the Automatic Database Diagnostic Monitor (ADDM) page.
   3) On the Automatic Database Diagnostic Monitor (ADDM) page, the ADDM Performance Analysis table shows you the consolidation of ADDM reports from all instances running in your cluster. This is your first entry point before drilling down to specific instances. From there, investigate the Buffer Busy findings. You should no longer see the Sequence Usage, nor specific instances impacted.
   4) The Global Cache Busy finding should not reveal anything special.

8. Close all terminal windows opened for this practice.

Practices for Lesson 8: Managing High Availability of Services

Chapter 8

Practices for Lesson 8: Overview

Practices Overview

In these practices, you will create, manage, and monitor services.

Practice 8-1: Working with Services

Overview

In this practice, you will use Enterprise Manager to create one service called `prod1`. You then observe what happens to your service when you terminate the instances on which it is running.

1. Use Enterprise Manager to create a singleton service called `prod1`.
   1) Open a terminal session from your PC desktop to host em12 as the oracle user. Enable *X* forwarding using `ssh` with the `-X` option. Start Firefox.

```
[vncuser@EDRSR46P1- ~]$ ssh -X oracle@em12
oracle@em12's password:
Last login: Tue Apr  9 12:58:10 2013 from 192.0.2.1
[oracle@em12 ~]$ firefox&
[1] 20289
[oracle@em12 ~]$
```

   Enter the Enterprise Manager URL in the browser you just started:

   `https://em12:7803/em`

   2) Log in using the following credentials: **`sysman/oracle_4U`**
   3) Expand the Targets menu and select All Targets. Scroll down the All Targets page and click orcl.
   4) Click the Availability pull-down menu and select the Cluster Managed Database Services option.
   5) On the Cluster Managed Database Services: Cluster and Database Login page, click the New radio button and provide the login credentials for the operating system user (`oracle/oracle`) and the SYSDBA credentials for the database (`sys/oracle_4U` as `sysdba`) and click Continue.
   6) Click the Create Service button on the Cluster Managed Database Services page.
   7) On the Create Service page, enter `prod1` for the service name. Verify that the “Start service after creation” check box is selected, and select the “Update local naming” check box. Under the High Availability Configuration section, select SINGLETON. Leave the remaining fields with their default values and click the OK button.
   8) After the service has been created, you will be returned to the Cluster Managed Database Services page. Check the Running Instances column for `prod1`, it should indicate the service running on one of the two instances. Select `prod1` from the Services list and click the Test Connection button. It should test successfully. Click the Show All TNS Strings button and inspect the new entry to the `tnsnames.ora` file. It should look like this:

```
prod1 = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST =
   cluster01-scan.cluster01.example.com)(PORT =
   1521))(LOAD_BALANCE = YES)(CONNECT_DATA = (SERVER =
   DEDICATED)(SERVICE_NAME = prod1)))
```

   9) Click the Return button.

2. Open a terminal session to `host01` as the `oracle` user. Set your environment with the `oraenv` script. Use the `srvctl` command to check the status of the new service. Take note on what host the service is currently running.

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

[oracle@host01 ~]$ srvctl status service -db ORCL -s prod1
Service prod1 is running on nodes: host02
[oracle@host01 ~]$
```

3. Use the `crsctl` command to view server pool relationships with the new service.

```
[oracle@host01 ~]$ /u01/app/12.1.0/grid/bin/crsctl status
serverpool -p
NAME=Free
IMPORTANCE=0
MIN_SIZE=0
MAX_SIZE=-1
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r-x
SERVER_CATEGORY=

NAME=Generic
IMPORTANCE=0
MIN_SIZE=0
MAX_SIZE=-1
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:grid:r-x,pgrp:oinstall:r-x,other::r-x
SERVER_CATEGORY=

NAME=ora.orcldb
IMPORTANCE=0
MIN_SIZE=0
MAX_SIZE=3
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:oracle:rwx,pgrp:oinstall:rwx,other::r--
```

```
SERVER_CATEGORY=ora.hub.category
[oracle@host01 ~]$
```

4. Connect to the service and look at the current value of the SERVICE_NAMES initialization parameter, and verify that it is set correctly. Query `V$INSTANCE` and determine what instance you are connected to.

```
[oracle@host01~]$ sqlplus sys/oracle_4U@prod1 as sysdba

SQL*Plus: Release 12.1.0.1.0 Production on Tue Sep 17 13:37:47
2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.



Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

SQL> show parameter service

NAME                 TYPE        VALUE
-------------------- ----------- ------------------------------
service_names           string orcl

SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
orcl_1

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
[oracle@host01~]$
```

5. From a terminal session on the node hosting the prod1 service, as the `oracle` user, crash the instance on the first node. In this example, the service is running on `host02`. Use ssh

to log into the host, find the database pmon process and kill the `ora_pmon_orcl_n` process. Use the `pkill -9 –f pmon_orcl1` command to crash the database instance. The `orcl1` instance will crash and the Clusterware services will restart it very quickly.

```
[oracle@host01 ~]$ ssh host02
Last login: Tue Sep 17 13:45:52 2013 from host01.example.com
[oracle@host02 ~]$ ps -ef|grep ora_pmon
oracle    4305     1  0 Sep13 ?        00:04:25 ora_pmon_orcl_2
oracle   26772 26746  0 13:47 pts/1    00:00:00 grep ora_pmon
[oracle@host02 ~]$ pkill -9 -f ora_pmon_orcl_2
[oracle@host02 ~]$
```

6. Use `srvctl` to check the status of the PROD1 service. (It may take a few moments to show up on the other host).

```
[oracle@host02 ~]$ . oraenv
ORACLE_SID = [oracl] ? orcl
The Oracle base has been set to /u01/app/oracle
[oracle@host02 ~]$ srvctl status service -db ORCL -s prod1
Service prod1 is running on nodes: host01
[oracle@host02 ~]$ exit
[oracle@host01 ~]$
```

7. Return to Enterprise Manager. Click the Availability link and select Cluster Managed Database Services. Click the `prod1` link. In the instance list under the Instances section, you should be able to verify that the first instance is indeed down.

8. Under Availability, click the Cluster Managed Database Services link. On the Cluster Managed Database Services page, you can see the current server hosting the service in the Running Servers column for `prod1`. Select Manage from the Actions drop-down list and click Go.

9. Find the server currently hosting the service and select the radio button for that host. Click the Relocate button.

10. On the Relocate Service from Instance page, select the host listed and click OK.

11. You should see a message indicating that the service was relocated successfully. You should see the service running on the original host.

Practice 8-2: Monitoring Services

Overview

In this practice, you will use Database Control to determine the amount of resources used by sessions executing under a particular service.

1. As the `oracle` user, open a terminal session to your first node. Execute the `/stage/RAC/labs/less_08/createuser.sh` script. This script creates a new user called `jmw` identified by the password `jmw`. The default tablespace of this user is `USERS`, and its temporary tablespace is `TEMP`. This new user has the `CONNECT`, `RESOURCE`, and `DBA` roles.

```
[oracle@host01 ~]cd /stage/RAC/labs/less_08

[oracle@host01 less_08]$ cat createuser.sh
export HOST=`hostname -s`
export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
export ORACLE_SID=`$ORACLE_HOME/bin/srvctl status database -db
orcl|grep $HOST|cut -f2 -d" "`
export PATH=$PATH:$ORACLE_HOME/bin


/u01/app/oracle/product/12.1.0/dbhome_1/bin/sqlplus -s /NOLOG
<<EOF

 connect / as sysdba
 drop user jmw cascade;
 create user jmw identified by jmw default tablespace users
temporary tablespace temp;
 grant connect, resource, dba to jmw;

EOF

[oracle@host01 less_08]$ ./createuser.sh

User dropped.


User created.


Grant succeeded.

[oracle@host01 less_08]$
```

2. Using SQL*Plus, connect to `prod1` as `jmw`. When connected, determine the instance on which your session is currently running. Then execute the following query:
   `select count(*) from dba_objects,dba_objects,dba_objects`
   Do not wait; instead, proceed with the next step..

```
$ sqlplus jmw/jmw@PROD1
SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
orcl_1

SQL> select count(*) from dba_objects,dba_objects,dba_objects;
```

3. In another terminal window as the oracle user, check statistics on your service with `gv$service_stats` from a SQL*Plus session connected as SYSDBA.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$ sqlplus sys/oracle_4U@orcl as sysdba

SQL*Plus: Release 12.1.0.1.0 Production on Tue Sep 17 14:47:31
2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.



Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

SQL> select stat_name, sum(value) from gv$service_stats where
service_name = 'prod1' group by stat_name;

STAT_NAME                                                SUM(VALUE)
-------------------------------------------------------- ----------
user calls                                                       18
DB CPU                                                    790706794
redo size                                                       704
db block changes                                                  4
DB time                                                   884881915
```

```
user rollbacks                                                    0
gc cr blocks received                                             2
gc cr block receive time                                          0
gc current blocks received                                        0
opened cursors cumulative                                       242
workarea executions - multipass                                   0

STAT_NAME                                                SUM(VALUE)
-------------------------------------------------------- ----------
session cursor cache hits                                       190
user I/O wait time                                           876444
parse count (total)                                             99
physical reads                                                  29
gc current block receive time                                    0
workarea executions - optimal                                    52
concurrency wait time                                         70511
parse time elapsed                                          2149526
physical writes                                                   0
workarea executions - onepass                                     0
execute count                                                   244

STAT_NAME                                                SUM(VALUE)
-------------------------------------------------------- ----------
session logical reads                                          7219
cluster wait time                                            875217
application wait time                                             0
logons cumulative                                                 1
sql execute elapsed time                                  884654643
user commits                                                      0

28 rows selected.

SQL>
```

4. Exit all SQL*Plus sessions and close all terminal windows opened for this practice.

```
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
[oracle@host01 ~]$

…

[oracle@host01 ~]$ exit
```

Note: If the `DBA_OBJECTS` query executed in step 2 has not finished at the end of this practice, enter <Ctl>-C in that terminal window to terminate the query.

Practice 8-3: Services and Alert Thresholds

Overview

In this practice, you will set thresholds for service PROD1, and use Database Control to monitor the response time metric for this service. In this practice, you will set the Elapsed Time in seconds warning threshold at 4 and the critical threshold at 1. Preferred instances should be orcl1 and orcl2, and orcl3 should be available.

1. Set alert thresholds for your service `prod1` using Enterprise Manager.
   1) Log in to Enterprise Manager as `sysman/oracle_4U` if you are not already logged in.
   2) On the Database Home page, click the Availability menu. Then click the Cluster Managed Database Services sub menu. On the Cluster Managed Database Services: Cluster and Database Login page, click the New radio button under Cluster Credentials and Database Credentials. Provide the login credentials for the operating system user (oracle/oracle) and the SYSDBA credentials for the database (sys/oracle_4U as sysdba) and click Continue.
   3) On the Cluster Database Home page, select Cluster Managed Database Services from the Availability pull-down menu. Note which host `prod1` is currently running on.
   4) Go to the Cluster Database home page, go down to the Instances section and click the instance link running on the server hosting the service.
   5) On the **orcl_orcl_*n*** Instance home page, select Monitoring from the Oracle Database pull-down menu, then select Metric and Collection Settings.
   6) On the Metric and Collection Settings page, select All metrics from the View pull down list.
   7) Scroll down the Metric and Collection Settings page until you find Service Response Time (per user call) (microseconds) metric located under Database Services.
   8) On the same line, click the corresponding multi-pens icon in the last column (Edit column).
   9) On the Edit Advanced Settings: Service Response Time (per user call) (microseconds) page, click Add.
   10) The Monitored Objects table should now show two entries.
   11) Enter `prod1` in the Service Name field, 40,000,000 in the Warning Threshold field, and 100,000,000 in the Critical Threshold field. Make sure that the corresponding line is selected, and click Continue.
   12) On the Metric and Policy Settings page, you should see an Information warning explaining that your settings have been modified but not saved. Click OK to save the new settings. If you see a Warning message at the top of the page regarding critical threshold values. Click OK again.
   13) On the Confirmation page, you can see an Update succeeded message. Click OK.
   14) This takes you back to the Database Instance page.

2. Use Enterprise Manager to view the Service Response Time Metric Value graphic for `prod1`.
   1) From the Instance home page, select Monitoring from the Oracle Database pull-down menu, then select All Metrics.

2) On the All Metrics page, expand the Database Services link, then click Service Response Time (per user call) (microseconds).
3) Select Real Time from the View Data pull-down menu.
4) You should now see configured services. Click the `prod1` column in the Service Name column.
5) Under the **Service Name: prod1** section, you should now see the Service Response Time (per user call) (microseconds) thresholds set with the values you previously entered.

3. Execute the `serv_wkload.sh` script to generate workload on your database. Looking at the Service Response time graphic for `prod1`, what do you observe?

```
[oracle@host01 less_08]$ ./serv_wkload.sh


[oracle@host01 less_08]$
```

1) Still looking at the Service Response Time (per user call) (microseconds): Service Name `prod1` page on your first session, you should see the graphic crossing the warning threshold after few minutes. This will trigger a warning alert soon after the warning threshold is crossed.
2) You can see this alert propagated to your Database Instance Home page, and Cluster Database Home page.
3) To go back to your Database Instance Home page, click the Database Instance locator link (orcl_orcl_*n*) on the Service Response Time page.
4) You should see the warning raised (Metrics Elapsed Time) in the Incidents and Problems section of the Database Instance page.
5) Navigate to the Cluster Database home page.
6) You should see the warning alert in the Incidents and Problems section of the page. Clicking the metric alert link takes you to the Incident Manager page. On the General tab, the alert is shown with its details. There are also tabs for Events, My Oracle Support Knowledge, Updates and Related Events and Incidents.
7) Soon after the script finishes its execution, you should not see the corresponding alert on your Cluster Database Home page anymore. You can navigate to the Alert History page found by selecting the Monitoring link on the Cluster Database pull-down menu.

4. Use Enterprise Manager to remove the thresholds that you specified during this practice.
   1) From the Cluster Database Home page, click the instance link corresponding to the server hosting the `prod1` service.
   2) On the Database Instance page, click the Oracle Database menu tab, select Monitoring, then select Metric and Collection Settings.
   3) On the Metric and Collection Settings page, scroll down the page until you see `prod1` under Database Services in the Metric Thresholds table.
   4) On the line corresponding to the `prod1` entry, remove both the Warning Threshold and Critical Threshold values.
   5) Click OK. You should see a Warning message at the top of the page regarding critical threshold values. Click OK again.
   6) On the Confirmation page, you should see an Update succeeded message. Click OK.
   7) Next, stop and remove the `prod1` service. Go to the Cluster database home page. Click Targets, select All Targets and click orcl. Click the Availability menu tab and select Cluster Managed Database services. If you are prompted to provide cluster and database credentials, click the New radio button and enter `oracle/oracle` for the cluster and `sys/oracle_4U` as `sysdba` for the database and click Continue. Click the prod1 radio button and then select delete from the Actions pull-down menu. Click Go. On the Delete Service: prod1 page, click yes.

5. Open a terminal session on your PC desktop and `su` to the `root` account. Shut down the `em12` host using the `xm shutdown` command. Monitor the process with the `xm list` command. Close the terminal window when finished.

```
[vncuser@classsroom_pc ~]$ su -
Password:
[root@classsroom_pc ~]# xm shutdown em12

[root@classsroom_pc ~]# xm list
Name                          ID   Mem VCPUs      State   Time(s)
Domain-0                       0  1044     2     r-----  21709.9
host01                        24  4200     1     -b----   1307.5
host02                        23  3200     2     -b---- 131536.0

[root@classsroom_pc ~]# exit

[vncuser@classsroom_pc ~]$ exit
```

6. Close all terminal windows opened for this practice.

Practices for Lesson 9: High Availability for Connections and Applications

Chapter 9

Practices for Lesson 9: Overview

Practices Overview

In this practice, you will explore Application Continuity.

Practice 9-1: Using Application Continuity

Overview

In this practice, you will use Application Continuity against a RAC database to demonstrate how Application Continuity helps an application to seamlessly recover after the failure of a RAC instance.

Tasks

1. Establish a terminal session connected to `host01` using the `oracle` OS user.

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

2. Configure the environment by using the `oraenv` script. Enter `orcl` when you are prompted for an ORACLE_SID value.

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

3. Confirm that two instances of the RAC database are up and running.

```
[oracle@host01 ~]$ srvctl status database -d orcl
Instance orcl_2 is running on node host02
Instance orcl_3 is running on node host01

[oracle@host01 ~]$
```

4. Navigate to the directory that contains the files for this practice.

```
[oracle@host01 ~]$ cd /stage/RAC/labs/less_09
[oracle@ host01 less_09]$
```

5. Create a database service on the `orcl` database. Configure the service for use in conjunction with Application Continuity.

```
[oracle@host01 less_09]$ srvctl add service -db orcl -service
actest -serverpool ora.orcldb -cardinality singleton -
failovertype TRANSACTION -commit_outcome TRUE -failoverretry 50
-failoverdelay 5 -retention 86400 -replay_init_time 1800 -
notification TRUE
[oracle@host01 less_09]$
```

6. Start the service.

```
[oracle@host01 less_09]$ srvctl start service -db orcl -service
actest
[oracle@host01 less_09]$
```

7. Examine the status of the newly created service. Take note of the node it is running on (`host02` in this case), because it may be different in your environment.

```
[oracle@host01 less_09]$ srvctl status service -db orcl -service
actest
Service actest is running on nodes: host02
[oracle@host01 less_09]$
```

8. Using SQL*Plus, connect to the `orcl` database as the `system` user on `host02`.

```
[oracle@host01 less_09]$ ssh host02

[oracle@host02 ~]$ cd /stage/RAC/labs/less_09

[oracle@host02 less_09]$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base has been set to /u01/app/oracle

[oracle@host02 less_09]$ sqlplus system/oracle_4U@orcl

SQL*Plus: Release 12.1.0.1.0 Production...

SQL>
```

9. Configure the `scott` database user as shown below. This is required for the application that you will use later. After the user is configured, exit your SQL*Plus session.

```
SQL> alter user scott identified by tiger account unlock;

User altered.

SQL> exit
Disconnected from Oracle Database 12c...
[oracle@host02 less_09]$
```

10. Establish another terminal session connected to `host02` using the `oracle` OS user. To differentiate this session from your primary session, it will be referred to as the ADMIN session for the rest of the practice.

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

[oracle@host02 ~]$ cd /stage/RAC/labs/less_09
[oracle@host02 less_09]$
```

11. Configure the prompt in your ADMIN session as shown below. This will help you to differentiate between your terminal sessions as you progress through this practice.

```
[oracle@host02 less_09]$ export PS1='ADMIN $ '
ADMIN $
```

12. Configure the environment in your ADMIN session by using the `oraenv` script. Enter `orcl` when you are prompted for an ORACLE_SID value.

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

13. Back in your first session, examine the scripts that you will soon use to execute the practice application. Notice that both scripts execute the same application code (in `actest.jar`). The only difference is that each script references a different properties file.

```
[oracle@host02 less_09]$ cat runnoreplay
java -classpath
./actest.jar:$ORACLE_HOME/ucp/lib/ucp.jar:$ORACLE_HOME/jdbc/lib/
ojdbc6.jar actest.ACTest actest_noreplay.properties
[oracle@host02 less_09]$
```

```
[oracle@host02 less_09]$ cat runreplay
java -classpath
./actest.jar:$ORACLE_HOME/ucp/lib/ucp.jar:$ORACLE_HOME/jdbc/lib/
ojdbc6.jar actest.ACTest actest_replay.properties
[oracle@host02 less_09]$
```

14. Examine the properties files. Notice that the only difference is the `datasource` specification.

```
[oracle@host02 less_09]$ cat actest_noreplay.properties
username=scott
password=tiger
autoCommit=false

# Use standard  12.1 datasource no replay
datasource=oracle.jdbc.pool.OracleDataSource

url=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=
cluster01-
scan.cluster01.example.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NA
ME=actest)))

# 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@host02 less_09]$
```

```
[oracle@host02 less_09]$ cat actest_replay.properties
username=scott
password=tiger
autoCommit=false

# Use new 12.1 replay datasource
datasource=oracle.jdbc.replay.OracleDataSourceImpl

url=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=
cluster01-
```

```
scan.cluster01.example.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NA
ME=actest)))

# 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@host02 less_09]$
```

```
[oracle@host02 less_09]$ diff actest_replay.properties
actest_noreplay.properties
5,6c5,6
< # Use new 12.1 replay datasource
< datasource=oracle.jdbc.replay.OracleDataSourceImpl
---
> # Use standard  12.1 datasource no replay
> datasource=oracle.jdbc.pool.OracleDataSource
[oracle@host02 less_09]$
```

Next, you will execute the practice Java application twice. Once without the benefit of Application Continuity, and once with Application Continuity enabled. Notice that you will execute the same application and the only difference is the JDBC data source that is used on each occasion. The source files containing the application code are contained in the `src` directory. Feel free to examine the application code if you like.

15. Execute the practice application without the benefit of Application Continuity. Notice that while the application runs, a periodic status message is displayed.

```
[oracle@host02 less_09]$ ./runnoreplay
#####################################################
Connecting to
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=clus
ter01-
scan.cluster01.example.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NA
ME=actest)))
```

```
  # of Threads             : 6
  UCP pool size            : 2
  Thread think time        : 20 ms
####################################################

2 active connections, avg response time from db 38 ms
2 active connections, avg response time from db 23 ms
```

16. While the application continues to execute in the primary window, return to your ADMIN session and remind yourself about which node is running the `actest` service. Then, abort the database instance running the `actest` service (`host02` in the example shown below). Ensure that you abort the instance on the node running the service and not the other database node.

```
ADMIN $ srvctl status service -d orcl -s actest
Service actest is running on nodes: host02
ADMIN $ srvctl stop instance -db orcl -node host02 -stopoption
ABORT -force
ADMIN $
```

17. Return to your primary window and you should see a series of errors caused by the aborting the database instance. This is typical of applications that do not use Application Continuity. Press Ctrl + C to abort the application.

```
...
        at
oracle.ucp.jdbc.oracle.OracleJDBCConnectionPool.borrowConnection
(OracleJDBCConnectionPool.java:1441)
        at
oracle.ucp.jdbc.oracle.OracleConnectionConnectionPool.borrowConn
ection(OracleConnectionConnectionPool.java:81)
        at
oracle.ucp.jdbc.PoolDataSourceImpl.getConnection(PoolDataSourceI
mpl.java:1027)
        ... 4 more
.Exception occurred while getting connection:
oracle.ucp.UniversalConnectionPoolException: Cannot get
Connection from Datasource: java.sql.SQLRecoverableException:
Listener refused the connection with the following error:
ORA-12514, TNS:listener does not currently know of service
requested in connect descriptor
 .
0 active connections, avg response time from db 150418377 ms
^C
[oracle@host02 less_09]$
```

18. Restart the aborted instance and confirm the both RAC database instances are up and running again.

```
[oracle@host02 less_09]$ srvctl start instance -d orcl -n host02
[oracle@host02 less_09]$ srvctl status database -d orcl
Instance orcl_1 is running on node host02
Instance orcl_2 is running on node host01
[oracle@host02 less_09]$
```

19. Reexamine the status of the `actest` service. You should observe that the service is running on a different node compared to what you observed earlier. This is because the service was migrated when you aborted the database instance earlier in the practice. Exit from `host02`, returning the terminal to `host01`.

```
[oracle@host02 less_09]$ srvctl status service -d orcl -s actest
Service actest is running on nodes: host01
[oracle@host02 less_09]$ exit
logout
Connection to host02 closed.

[oracle@host01 less_09]$
```

20. Execute the practice application with Application Continuity enabled. You should see the same period status messages as before while the application is running.

```
[oracle@host01 less_09]$ ./runreplay
####################################################
Connecting to
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=clus
ter01-
scan.cluster01.example.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NA
ME=actest)))
  # of Threads             : 6
  UCP pool size            : 2
  Thread think time        : 20 ms
####################################################

2 active connections, avg response time from db 44 ms
2 active connections, avg response time from db 23 ms
2 active connections, avg response time from db 20 ms
```

21. While the application continues to execute in the primary window, return to your ADMIN session and remind yourself about which node is now running the `actest` service. Then, abort the database instance running the `actest` service (which is now `host01` in the example shown below). Ensure that you abort the instance on the node running the service and not the other database node.

```
ADMIN $ srvctl status service -d orcl -s actest
Service actest is running on nodes: host01
```

```
ADMIN $ srvctl stop instance -db orcl -node host01 -stopoption
ABORT -force
ADMIN $
```

22. Return to your primary window and you should see that the application continued in spite of aborted database instance. You should see a brief spike in the response time, which coincides with the time when the database instance was aborted. Now you have seen how Application Continuity masks the effect of database instance loss in a RAC database environment. Press Ctrl + C to abort the application.

```
[oracle@host01 less_09]$ ./runreplay
####################################################
Connecting to
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=clus
ter01-
scan.cluster01.example.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NA
ME=actest.cluster01.example.com)))
  # of Threads             : 6
  UCP pool size            : 2
  Thread think time        : 20 ms
####################################################

2 active connections, avg response time from db 44 ms
2 active connections, avg response time from db 23 ms
2 active connections, avg response time from db 20 ms
2 active connections, avg response time from db 16 ms
2 active connections, avg response time from db 14 ms
2 active connections, avg response time from db 12 ms
2 active connections, avg response time from db 12 ms
2 active connections, avg response time from db 15 ms
2 active connections, avg response time from db 425 ms
2 active connections, avg response time from db 16 ms
2 active connections, avg response time from db 13 ms
2 active connections, avg response time from db 12 ms
2 active connections, avg response time from db 12 ms
2 active connections, avg response time from db 13 ms
^C
[oracle@host01 less_09]$
```

23. Confirm that the database instance aborted and that the service migrated to the other node as expected.

```
[oracle@host01 less_09]$ srvctl status database -d orcl
Instance orcl_1 is running on node host02
Instance orcl_2 is not running on node host01
```

```
[oracle@host01 less_09]$ srvctl status service -d orcl -s actest
Service actest is running on nodes: host02
[oracle@host01 less_09]$
```

24. Restart the stopped database instance. Stop the `actest` service and remove it.

```
[oracle@host01 less_09]$ srvctl start instance -d orcl -n host01

[oracle@host01 less_09]$ srvctl stop service -d orcl -s actest

[oracle@host01 less_09]$ srvctl remove service -d orcl -s actest
[oracle@host01 less_09]$
```

25. Close all terminal windows opened for this practice.

Practices for Lesson 10: Upgrading and Patching Oracle RAC

Chapter 10

Practices for Lesson 10

Practices Overview

There are no practices for this lesson.

Practices for Lesson 11: Oracle RAC One Node

Chapter 11

Practices for Lesson 11: Overview

Practices Overview

In these practices, you will create a RAC One Node Database.

Practice 11-1: RAC One Node

Overview

In this practice, you will create a RAC One Node database. You will perform an online database relocation. Finally, you will convert the RAC One Database to an Oracle RAC database.

1. From a terminal session on your classroom PC, `su` to the `root` account and start `host03`. Next, , remove the existing RAC database. From your classroom PC desktop, execute `ssh –X oracle@host01` to open a terminal session on `host01` as the `oracle` user. Then navigate to /`u01/app/oracle/product/12.1.0/dbhome_1/bin` and execute DBCA.

```
[vncuser@EDRSR46P1 ~]$ su -
Password:
[root@EDRSR46P1 ~]# xm create host03
Using config file "/etc/xen/host03".
Started domain host03 (id=50)

[root@EDRSR46P1 ~]# ssh -X oracle@host01
oracle@host01's password:

***** Wait a few minutes for host03 to start *****

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

[oraclec@host01 ~]# cd
/u01/app/oracle/product/12.1.0/dbhome_1/bin

[oracle@host01 bin]$ ./dbca
```

Step	Screen/Page Description	Choices or Values
a.	Database Operation	Select Delete Database. Click Next.
b.	Delete Database	Select orcl and click Next.
c.	Management Options	Click Next.
d.	Summary	Click Finish.
e.	Database Configuration Assistant dialog box	You are informed that the instances and datafiles will be deleted. Click Yes to proceed.
f.	Database Configuration Assistant dialog box	You are informed that database deletion is complete. Click OK.
g.	Progress Page	Click Close.

2. To add the RAC One Node database, start DBCA again.

```
[oracle@host01 bin]$ ./dbca
```

Step	Screen/Page Description	Choices or Values
a.	Database Operations	Select Create Database. Click Next.
b.	Creation Mode	Select Advanced Mode. Click Next.
c.	Database Templates	Select Oracle RAC One Node database as the Database Type. Select Policy-Managed as the Configuration Type. Select the General Purpose or Transaction Processing template. Click Next.
d.	Database Identification	Enter **orcl** for the Global Database Name. Enter **serv1** for the Service Name. Click Next.
e.	Database Placement	Select Use Existing Server pool for this database, then select the orcldb server pool. Click Next.
f.	Management options	Un-select all options on the page and click Next.
g.	Database Credentials	Select "Use the same Administrative password." Enter oracle_4U as the password. Enter it again to confirm. Click Next.
h.	Storage Locations	In the Database Files section, select Automatic Storage Management (ASM) as the Storage Type. Select Oracle-Managed Files and enter +DATA in the Database File Locations field. In the Recovery Related Files section, select Automatic Storage Management (ASM) as the Storage Type. Select Specify Fast Recovery Area and enter +FRA in the Fast Recovery Area field. Accept the default value for Fast Recovery Area Size. Click Next.
i.	Database Options	Click Next.
j.	Initialization Parameters	Change memory Size (SGA and PGA) to 800 MB. Make sure that the Typical radio button is selected. Click the Character Sets tab and select Use Unicode (AL32UTF). Click Next.
k.	Creation Options	Select Create Database and click Next.
l.	Summary	Click Finish.
m.	Dialog box	Upon database completion, a dialog box is displayed. Click Exit.
n.	Progress Page	Click Close.

3. From the `oracle` terminal session, check your database configuration using the `srvctl` utility.

```
[oracle@host01 bin]$ srvctl config database -db orcl
Database unique name: orcl
Database name: orcl
Oracle home: /u01/app/oracle/product/12.1.0/dbhome_1
Oracle user: oracle
Spfile: +DATA/orcl/spfileorcl.ora
Password file: +DATA/orcl/orapworcl
```

```
Domain: cluster01.oracle.com
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Server pools: orcldb
Database instances:
Disk Groups: DATA
Mount point paths:
Services: serv1
Type: RACOneNode
Online relocation timeout: 30
Instance name prefix: orcl
Candidate servers:
Database is policy managed

[oracle@host01 bin]$
```

4. Use the `srvctl` utility to check the status of the `orcl` database.

```
[oracle@host01 bin]$ srvctl status database -db orcl
Instance orcl_1 is running on node host01
Online relocation: INACTIVE
[oracle@host01 bin]$
```

5. Execute `srvctl relocate database –help` to view command usage.

```
[oracle@host01 bin]$ srvctl relocate database -help

Initiate online relocation of the RAC One Node database.

Usage: srvctl relocate database -db <db_unique_name> {[-node
<target>] [-timeout <timeout>] | -abort [-revert]} [-verbose]
    -db <db_unique_name>           Unique name of database to
relocate
    -node <target>                 Target node to which to
relocate database
    -timeout <timeout>             Online relocation timeout in
minutes
    -abort                         Abort failed online
relocation
    -revert                        Remove target node of failed
online relocation request from the candidate server list of
administrator-managed RAC One Node database
    -verbose                       Verbose output
```

```
    -help                          Print usage
[oracle@host01 bin]$
```

6. Use `srvctl` to perform an online database relocation from host01 to host02.
   Immediately after issuing the command, proceed to the next step!

```
[oracle@host01 ~]$ srvctl relocate database -db orcl -node
host02 -w 15 -v

<<< Immediately go to the next step>>>

Configuration updated to two instances
Instance orcl_2 started
Services relocated
Waiting for up to 15 minutes for instance orcl_1 to stop ...
Instance orcl_1 stopped
Configuration updated to one instance

[oracle@host01 ~]$
```

7. Open another terminal window as `oracle`, set the environment and issue the `srvctl status database -db orcl` command several times to monitor the migration process.

```
[oracle@host01 ~]$ srvctl status database -db orcl
Instance orcl_2 is running on node host02
Online relocation: ACTIVE
Source instance: orcl_1 on host01
Destination instance: orcl_2 on host02

[oracle@host01 ~]$ srvctl status database -db orcl
Instance orcl_2 is running on node host02
Online relocation: ACTIVE
Source instance: orcl_1 on host01
Destination instance: orcl_2 on host02

[oracle@host01 ~]$ srvctl status database -db orcl
Instance orcl_2 is running on node host02
Online relocation: INACTIVE

[oracle@host01 ~]$
```

Make sure that instance orcl_2 is running on host02 before continuing to the next step

8. Let's convert our RAC One Node database to a RAC database. First, shut down the RAC One Node database.

```
[oracle@host01 ~]$ srvctl stop database -db orcl
[oracle@host01 ~]$
```

9. Use `srvctl` to convert the database to RAC and restart the database.

```
[oracle@host01 ~]$ srvctl convert database -db orcl -dbtype RAC

[oracle@host01]$ srvctl start database -db orcl
```

10. Execute the `srvctl status service` command to view the services configuration. Note the `serv1` service is running on all nodes.

```
[oracle@host01 ~]$ srvctl status service -d orcl
Service serv1 is running on nodes: host03,host01,host02

[oracle@host01 ~]$ srvctl stop service -db orcl -service serv1
[oracle@host01 ~]$ srvctl remove service -db orcl -service serv1
[oracle@host01 ~]$
```

11. Execute the `srvctl config database` command to view the database configuration.

```
[oracle@host01 ~]$ srvctl config database -d orcl
Database unique name: orcl
Database name: orcl
Oracle home: /u01/app/oracle/product/12.1.0/dbhome_1
Oracle user: oracle
Spfile: +DATA/orcl/spfileorcl.ora
Password file: +DATA/orcl/orapworcl
Domain:
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Server pools: orcldb
Database instances:
Disk Groups: DATA
Mount point paths:
Services: serv1
Type: RAC
Start concurrency:
Stop concurrency:
Database is policy managed
[oracle@host01 ~]$
```

12. Exit all terminal windows opened for this practice.

Practices for Lesson 12: Quality of Service Management

Chapter 12

Practices for Lesson 12

Practices Overview

There are no practices for this lesson.

Practices for Lesson 13: Multitenant Architecture and RAC Environment

Chapter 13

Practices for Lesson 13: Overview

Overview

In this practice, you will create a new CDB named `cdb1` including one PDB named `pdb1`. The CDB is hosted in an existing server pool.

Then you will create another PDB named `pdb2` and manage the services to affinitize the PDB services to instances.

At the end of the practice, you drop the `pdb2` PDB.

Practice 13-1: Creating a CDB

Overview

In this practice, you will create a new CDB named `cdb1` with DBCA.

Pre CDB Creation Tasks

As `root`, start `host03`. Then, remove the existing RAC database. From your classroom PC desktop, execute `ssh –X oracle@host01` to open a terminal session on host01 as the oracle user. Then navigate to /`u01/app/oracle/product/12.1.0/dbhome_1/bin` and execute DBCA.

```
[root@EDRSR46P1 ~]# ssh -X oracle@host01
oracle@host01's password:

[oraclec@host01 ~]# cd
/u01/app/oracle/product/12.1.0/dbhome_1/bin

[oracle@host01 bin]$ ./dbca
```

Step	Screen/Page Description	Choices or Values
a.	Database Operation	Select Delete Database. Click Next.
b.	Delete Database	Select orcl and click Next.
c.	Management Options	Click Next.
d.	Summary	Click Finish.
e.	Database Configuration Assistant dialog box	You are informed that the instances and datafiles will be deleted. Click Yes to proceed.
f.	Database Configuration Assistant dialog box	You are informed that database deletion is complete. Click OK.
g.	Progress Page	Click Close.

Remove the `orcldb` server pool..

```
[oracle@host01 bin]$ /u01/app/12.1.0/grid/bin/srvctl remove
srvpool -serverpool orcldb
[oracle@host01 bin]$ /u01/app/12.1.0/grid/bin/srvctl status
srvpool
Server pool name: Free
Active servers count: 3
Server pool name: Generic
Active servers count: 0
[oracle@host01 bin]$
```

Tasks

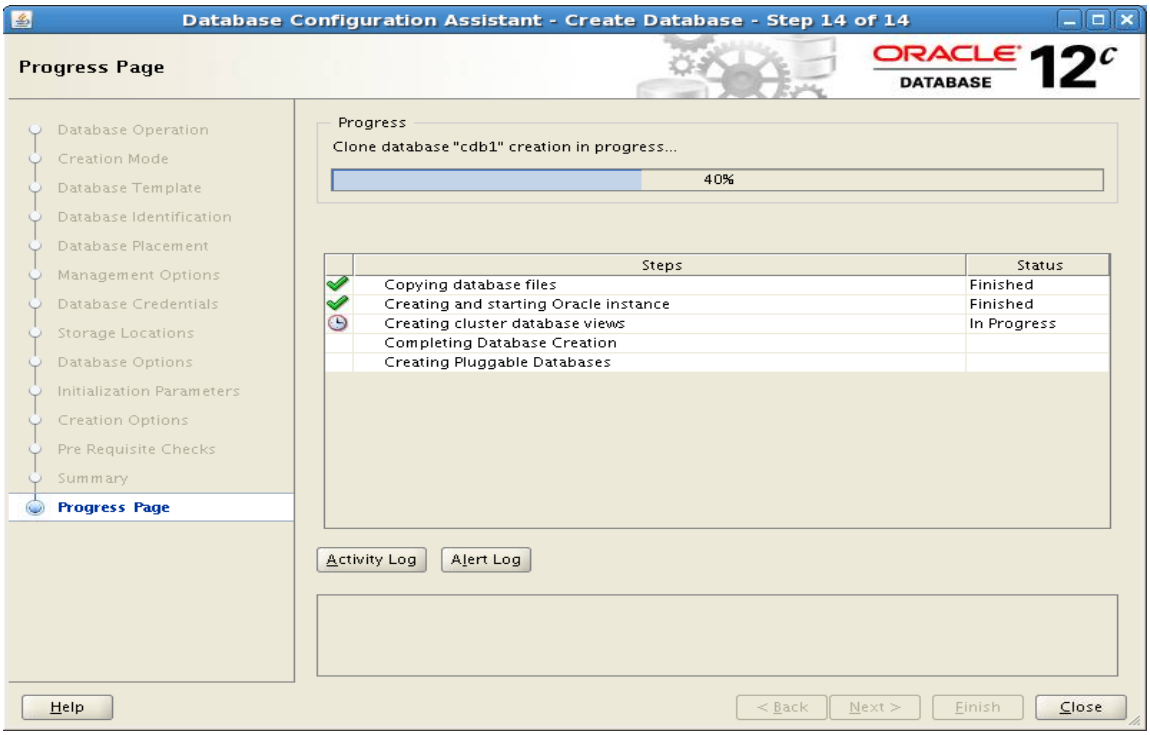
1. Start DBCA and perform the following steps.

```
[oracle@host01 bin]$ ./dbca
```

Step	Window/Page Description	Choices or Values
a.	Step 1: Database Operation	Select “Create Database.” Click Next.
b.	Step 2: Creation Mode	Select “Advanced Mode.” Click Next.
c.	Step 3: Database Template	Select “Oracle Real Application Clusters (RAC) database” for Database Type. Select “Policy-Managed” for Configuration Type. Select “General Purpose or Transaction Processing.” Click Next.
d.	Step 4: Database Identification	Enter Global Database Name: `cdb1` Select **“Create As Container Database.”** Select **“Create A Container Database with one or more PDBs.**” Select 1 for Number of PDBs. Enter `pdb1` for PDB Name. Click Next.
e.	Step 5: Database Placement	Select “Create New Server pool for this database”. Enter “`cdb1pool`“ for Server pool Name and 3 for Cardinality. Click Next.
f.	Step 6: Management Options	Deselect “Configure Enterprise Manager (EM) Database Express.” Click Next.
g.	Step 7: Database Credentials	Select “Use same Administrative password…” Enter: Password: `oracle_4U` Confirm password: `oracle_4U` Click Next.
h.	Step 8: Storage Locations	Confirm Storage type is “Automatic Storage Management (ASM).” Confirm “Use Common Location for All Database Files.” in `+DATA` diskgroup. Deselect “Specify Fast Recovery Area”. Click Next.
i.	Step 9: Database Options	Click Next.
j.	Step 10: Initialization Parameters	Set “Memory Size (SGA and PGA)” to 840 MB.

Step	Window/Page Description	Choices or Values
		Select “Use Automatic Memory Management” Select “Character Sets.” Select “Use Unicode (**AL32UTF8**).” Click Next.
k.	Step 11: Creation Option	Select “Create Database.” Click Next.
l.	Step 12: Pre Requisite Checks	Click Next.
m.	Step 13: Summary	Click Finish.
n.	Step 14: Progress Page	On the Database Configuration Assistant page (for password management), click Exit. Click Close.

The screenshot below corresponds to step n.



2. Explore the CDB instances hosted on the nodes of the server pool. You will also see that the `pdb1` PDB can be accessed on any instance of the CDB just like a non-CDB can be accessed on any instance in a RAC environment.
   a. Check the `cdb1pool` server pool and its cardinality.

```
[oracle@host01 bin]$ su - grid
grid@host01's password:
[grid@host01 ~]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base has been set to /u01/app/grid
```

```
[grid@host01 ~]$ srvctl status srvpool
Server pool name: Free
Active servers count: 0
Server pool name: Generic
Active servers count: 0
Server pool name: cdb1pool
Active servers count: 3

[grid@host01 ~]$ srvctl status srvpool -serverpool cdb1pool
Server pool name: cdb1pool
Active servers count: 3

[grid@host01 ~]$ exit
logout
Connection to host01 closed.
[oracle@host01 ~]$
```

b. Use SRVCTL to know on which nodes the instances of the CDB are running, as you traditionally do for any non-CDB.

```
[oracle@host01 ~]$ export
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
[oracle@host01 ~]$ cd $ORACLE_HOME/bin

[oracle@host01 bin]$ ./srvctl status database -d cdb1
Instance cdb1_1 is running on node host03
Instance cdb1_2 is running on node host02
Instance cdb1_3 is running on node host01

[oracle@host01 bin]$ pgrep -l cdb1_3
9770 ora_pmon_cdb1_3
9772 ora_psp0_cdb1_3
9774 ora_vktm_cdb1_3
9778 ora_gen0_cdb1_3
9780 ora_mman_cdb1_3
9784 ora_diag_cdb1_3
9786 ora_dbrm_cdb1_3
9790 ora_ping_cdb1_3
9792 ora_acms_cdb1_3
9794 ora_dia0_cdb1_3
9796 ora_lmon_cdb1_3
9798 ora_lmd0_cdb1_3
```

```
9800 ora_lms0_cdb1_3
9804 ora_rms0_cdb1_3
9806 ora_lmhb_cdb1_3
9808 ora_lck1_cdb1_3
9810 ora_dbw0_cdb1_3
9812 ora_lgwr_cdb1_3
9814 ora_ckpt_cdb1_3
9816 ora_smon_cdb1_3
9818 ora_reco_cdb1_3
9820 ora_lreg_cdb1_3
9822 ora_rbal_cdb1_3
9824 ora_asmb_cdb1_3
9826 ora_mmon_cdb1_3
9830 ora_mmnl_cdb1_3
9832 ora_d000_cdb1_3
9834 ora_s000_cdb1_3
9836 ora_mark_cdb1_3
9841 ora_gcr0_cdb1_3
9843 ora_lck0_cdb1_3
9857 ora_rsmn_cdb1_3
9906 ora_tmon_cdb1_3
9908 ora_tt00_cdb1_3
9960 ora_smco_cdb1_3
9962 ora_w000_cdb1_3
9968 ora_gtx0_cdb1_3
9970 ora_rcbg_cdb1_3
9972 ora_ppa7_cdb1_3
9987 ora_aqpc_cdb1_3
9989 ora_qm02_cdb1_3
9991 ora_q001_cdb1_3
9993 ora_q002_cdb1_3
9995 ora_qm05_cdb1_3
10013 ora_p000_cdb1_3
10015 ora_p001_cdb1_3
10017 ora_p002_cdb1_3
10019 ora_p003_cdb1_3
10319 ora_cjq0_cdb1_3
11166 ora_w001_cdb1_3
11995 ora_w002_cdb1_3
13641 ora_p004_cdb1_3
13643 ora_p005_cdb1_3
[oracle@host01 bin]$
```

c. Use LSNRCTL to list the CDB instances on two nodes of the server pool.

1) Check the services on the first node.

```
[oracle@host01 bin]$ ./lsnrctl status

LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 02-SEP-
2013 06:54:40

Copyright (c) 1991, 2013, Oracle.  All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.1.0
- Production
Start Date                30-AUG-2013 07:22:00
Uptime                    2 days 23 hr. 32 min. 40 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/12.1.0/grid/network/admin/listener.ora
Listener Log File
/u01/app/grid/diag/tnslsnr/host01/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.247)(PORT=1521
)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.101)(PORT=1521
)))
Services Summary...
Service "+ASM" has 1 instance(s).
  Instance "+ASM1", status READY, has 1 handler(s) for this
service...
Service "cdb1" has 1 instance(s).
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
Service "cdb1XDB" has 1 instance(s).
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
Service "pdb1" has 1 instance(s).
```

```
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
The command completed successfully
[oracle@host01 bin]$
```

2) Check the services on the second node.

```
[oracle@host01 bin]$ ssh host02
Last login: Mon Sep  2 01:17:02 2013 from 192.0.2.1
[oracle@host02 ~]$ . oraenv
ORACLE_SID = [oracle] ? cdb1_2
ORACLE_HOME = [/home/oracle] ?
/u01/app/oracle/product/12.1.0/dbhome_1
The Oracle base has been set to /u01/app/oracle

[oracle@host02 ~]$ lsnrctl status
…
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.245)(PORT=1521
)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.102)(PORT=1521
)))
Services Summary...
Service "+ASM" has 1 instance(s).
  Instance "+ASM3", status READY, has 1 handler(s) for this
service...
Service "cdb1" has 1 instance(s).
  Instance "cdb1_2", status READY, has 1 handler(s) for this
service...
Service "cdb1XDB" has 1 instance(s).
  Instance "cdb1_2", status READY, has 1 handler(s) for this
service...
Service "pdb1" has 1 instance(s).
  Instance "cdb1_2", status READY, has 1 handler(s) for this
service...
The command completed successfully

[oracle@host02 ~]$ exit
logout
Connection to host02 closed.
[oracle@host01 bin]$
```

d. Use SRVCTL to stop and restart the CDB as you traditionally would do for any non-CDB.

```
[oracle@host01 bin]$ . oraenv
ORACLE_SID = [oracle] ? cdb1_3
ORACLE_HOME = [/home/oracle] ?
/u01/app/oracle/product/12.1.0/dbhome_1
The Oracle base has been set to /u01/app/oracle

[oracle@host01 bin]$ srvctl stop database -d cdb1
[oracle@host01 bin]$ srvctl status database -db cdb1
Instance cdb1_1 is not running on node host03
Instance cdb1_2 is not running on node host02
Instance cdb1_3 is not running on node host01

[oracle@host01 bin]$ srvctl start database -d cdb1

[oracle@host01 bin]$ srvctl status database -db cdb1
Instance cdb1_1 is running on node host03
Instance cdb1_2 is running on node host02
Instance cdb1_3 is running on node host01

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

e. Use SQL*Plus to connect to the instances of the `cdb1` CDB, check the UNDO tablespaces and the groups of redo log files, and verify the existence of the `pdb1` PDB.

1) Check the UNDO tablespaces created in the CDB.

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

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP, Advanced Analytics and Real
Application Testing options

SQL> SELECT name, cdb, con_id FROM v$database;

NAME      CDB     CON_ID
--------- --- ----------
CDB1      YES          0

SQL> SELECT instance_name, con_id FROM v$instance;
```

```
INSTANCE_NAME        CON_ID
---------------- ----------
cdb1_3                    0

SQL> show con_name

CON_NAME
------------------------------
CDB$ROOT

SQL> SELECT tablespace_name, con_id
     FROM   cdb_tablespaces
     WHERE  contents = 'UNDO';

TABLESPACE_NAME                   CON_ID
------------------------------ ----------
UNDOTBS1                               1
UNDOTBS2                               1
UNDOTBS3                               1

SQL>
```

2) Check the groups of redo log files created for the three CDB instances.

```
SQL> SELECT group#, con_id FROM   v$logfile;

    GROUP#     CON_ID
---------- ----------
         2          0
         1          0
         5          0
         6          0
         3          0
         4          0

6 rows selected.

SQL>
```

3) Check the PDB created in the CDB and its open mode. If the PDB is not opened, open it.

```
SQL> COL pdb_name format a10
```

```
SQL> SELECT pdb_id, pdb_name, guid, status FROM cdb_pdbs;

    PDB_ID PDB_NAME   GUID                             STATUS
---------- ---------- -------------------------------- --------
         3 PDB1       E13E44A728D5266BE043650200C0187D NORMAL
         2 PDB$SEED   E13D83F6E4966F2AE043650200C0058C NORMAL

SQL> SELECT name, open_mode FROM v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB1                           MOUNTED

SQL> ALTER SESSION SET CONTAINER=pdb1;

Session altered.

SQL> show con_name

CON_NAME
------------------------------
PDB1

SQL> CONNECT / AS SYSDBA
Connected.

SQL> SELECT name FROM cdb_services;

NAME
----------------------------------------------------------------
SYS$BACKGROUND
SYS$USERS
cdb1XDB
cdb1

SQL> ALTER PLUGGABLE DATABASE pdb1 OPEN;

Pluggable database altered.

SQL> SELECT name, open_mode FROM v$pdbs;
```

```
NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB1                           READ WRITE

SQL>
```

4) Check the services.

```
SQL> SELECT name FROM v$services;

NAME
----------------------------------------------------------------
pdb1
cdb1XDB
cdb1
SYS$BACKGROUND
SYS$USERS

SQL> EXIT
```

f. Switch to the second node to verify the open mode of the PDB in the second instance of the CDB.

```
[oracle@host01 ~]$ ssh host02
[oracle@host02 ~]$ . oraenv
ORACLE_SID = [oracle] ? cdb1_2
ORACLE_HOME = [/home/oracle] ?
/u01/app/oracle/product/12.1.0/dbhome_1
The Oracle base has been set to /u01/app/oracle

[oracle@host02 ~]$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP, Advanced Analytics and Real
Application Testing options

SQL> SELECT name, cdb, con_id FROM v$database;

NAME      CDB     CON_ID
--------- --- ----------
CDB1      YES          0
```

```
SQL> SELECT instance_name, con_id FROM v$instance;

INSTANCE_NAME        CON_ID
---------------- ----------
cdb1_2                    0

SQL> show con_name

CON_NAME
------------------------------
CDB$ROOT

SQL> SELECT name, open_mode FROM v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB1                           MOUNTED

SQL> ALTER SESSION SET CONTAINER=pdb1;

Session altered.

SQL> SELECT name FROM v$services;

NAME
----------------------------------------------------------------
pdb1

SQL> exit
[oracle@host02 ~]$ exit
logout
Connection to host02 closed.
[oracle@host01 ~]$
```

g. Verify that the `pdb1` service is accessible from instance `cdb1_3` on the first node but also from `cdb1_2` instance on the second node and from `cdb1_1` instance on the third node. First restart the listener.

```
[oracle@host01 ~]$ su - grid
Password:
Last login: Mon Sep  2 05:16:31 2013 from host01.example.com
```

```
[grid@host01 ~]$  . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base has been set to /u01/app/grid

[grid@host01 ~]$ srvctl stop listener -listener LISTENER

[grid@host01 ~]$ srvctl start listener -listener LISTENER

[grid@host01 ~]$ exit
logout
Connection to host01 closed.

[oracle@host01 ~]$ sqlplus /nolog
SQL*Plus: Release 12.1.0.1.0 Production on Wed Jul 17 00:38:28
2013

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> CONNECT system@"host01:1521/pdb1"
Enter password:
Connected.
SQL> SELECT instance_name, con_id FROM v$instance;

INSTANCE_NAME        CON_ID
---------------- ----------
cdb1_3                    0

SQL> show con_name

CON_NAME
------------------------------
PDB1
SQL> CONNECT system@"host02:1521/pdb1"
Enter password:
ERROR:
ORA-01033:ORACLE initialization or shutdown in progress
Process ID: 0
Session ID: 0 Serial Number: 0

Warning: You are no longer connected to ORACLE.
SQL>
```

Notice that the connection does not complete because `pdb1` was opened for instance `cdb1_3` on `host01` only. Remember that the clause `INSTANCES` was not used in the `ALTER PLUGGABLE DATABASE OPEN` statement in task 4.e.3).

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> ALTER PLUGGABLE DATABASE pdb1 OPEN INSTANCES=('cdb1_2');

Pluggable database altered.
```

```
SQL> CONNECT system@"host02:1521/pdb1"
Enter password:
Connected.

SQL> SELECT instance_name, con_id FROM v$instance;

INSTANCE_NAME        CON_ID
---------------- ----------
cdb1_2                    0

SQL> show con_name

CON_NAME
------------------------------
PDB1

SQL> CONNECT system@"host03:1521/pdb1"
Enter password:
ERROR:
ORA-01033:ORACLE initialization or shutdown in progress
Process ID: 0
Session ID: 0 Serial Number: 0

Warning: You are no longer connected to ORACLE.

SQL> EXIT
```

The connection does not complete on host03 because `pdb1` was opened for instance `cdb1_3` on `host01` and `cdb1_2` on `host02` only.

Practice 13-2: Cloning a PDB in the RAC CDB

Overview

In this practice, you will clone the `pdb1` PDB into a new PDB named `pdb2` in the `cdb1` CDB. This operation requires to close and open PDBs on multiple instances of the CDB.

Tasks

1. Connect to the root of the multitenant container database `cdb1` on any of the three instances.

```
[oracle@host01 ~]$ . oraenv
ORACLE_SID = [cdb1] ? cdb1_3
ORACLE_HOME = [/home/oracle] ?
/u01/app/oracle/product/12.1.0/dbhome_1
The Oracle base has been set to /u01/app/oracle
[oracle@host01 ~]$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL>
```

2. Use Oracle Managed Files to locate the data files of the new `pdb2`.

```
SQL> SHOW PARAMETER db_create_file_dest
NAME                                 TYPE        VALUE
------------------------------------ ----------- --------------
db_create_file_dest                  string      +DATA

SQL>
```

3. Create `pdb2` from `pdb1`.
   a. Use the `CREATE PLUGGABLE DATABASE` command to create `pdb2`.

```
SQL> CREATE PLUGGABLE DATABASE pdb2 FROM pdb1;
CREATE PLUGGABLE DATABASE pdb2 FROM pdb1
*
ERROR at line 1:
ORA-65081: database or pluggable database is not open in read
only mode

SQL> SELECT name, open_mode FROM v$pdbs;
```

```
NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB1                           READ WRITE

SQL> ALTER PLUGGABLE DATABASE pdb1 CLOSE IMMEDIATE;

Pluggable database altered.

SQL> ALTER PLUGGABLE DATABASE pdb1 OPEN READ ONLY;
ALTER PLUGGABLE DATABASE pdb1 OPEN READ ONLY
*
ERROR at line 1:
ORA-16002: database or pluggable database already open for
read/write access by another instance

SQL>
```

b. Close pdb1 in all the CDB instances, not in the current instance only because other connections to pdb1 can be performed in the other two CDB instances.

```
SQL> ALTER PLUGGABLE DATABASE pdb1 CLOSE IMMEDIATE
INSTANCES=ALL;

Pluggable database altered.

SQL> ALTER PLUGGABLE DATABASE pdb1 OPEN READ ONLY;

Pluggable database altered.

SQL> CREATE PLUGGABLE DATABASE pdb2 FROM pdb1;

Pluggable database created.

SQL> SELECT name, open_mode FROM v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB1                           READ ONLY
PDB2                           MOUNTED

SQL> COL pdb_name format a10
SQL> SELECT pdb_id,pdb_name, guid, status FROM cdb_pdbs;
```

```
    PDB_ID PDB_NAME   GUID                                STATUS
---------- ---------- ---------------------------------- --------
         3 PDB1       E13E44A728D5266BE043650200C0187D NORMAL
         2 PDB$SEED   E13D83F6E4966F2AE043650200C0058C NORMAL
         4 PDB2       E2B1483E90856557E043650200C01D40 NEW

SQL>
```

c. Now, open both PDBs in READ WRITE mode on all the CDB instances.

```
SQL> ALTER PLUGGABLE DATABASE pdb1 CLOSE IMMEDIATE
INSTANCES=ALL;

Pluggable database altered.

SQL> ALTER PLUGGABLE DATABASE ALL OPEN READ WRITE INSTANCES=ALL;

Pluggable database altered.

SQL> SELECT name, open_mode FROM v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB1                           READ WRITE
PDB2                           READ WRITE

SQL> SELECT pdb_id, pdb_name, guid, status FROM cdb_pdbs;

    PDB_ID PDB_NAME   GUID                                STATUS
---------- ---------- ---------------------------------- --------
         3 PDB1       E13E44A728D5266BE043650200C0187D NORMAL
         2 PDB$SEED   E13D83F6E4966F2AE043650200C0058C NORMAL
         4 PDB2       E2B1483E90856557E043650200C01D40 NORMAL

SQL> ALTER SESSION SET CONTAINER=pdb2;

Session altered.

SQL> SELECT name FROM dba_services;

NAME
```

```
-------------------------------------------------------------------
pdb2

SQL> EXIT
```

d. Use LSNRCTL to verify that the new `pdb2` service associated to the new PDB in the CDB instance is automatically started after the PDB is opened. Because the PDB is opened in all the CDB instances, the `pdb2` PDB service is started in all the CDB instances.

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

LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 30-JUL-
2013 05:44:45

Copyright (c) 1991, 2013, Oracle.  All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.1.0
- Production
Start Date                16-JUL-2013 05:38:51
Uptime                    14 days 0 hr. 5 min. 55 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listening Endpoints Summary...
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.101)(PORT=1521
)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.239)(PORT=1521
)))
Services Summary...
Service "+APX" has 1 instance(s).
  Instance "+APX1", status READY, has 1 handler(s) for this
service...
Service "+ASM" has 1 instance(s).
  Instance "+ASM1", status READY, has 2 handler(s) for this
service...
Service "DBUA0954399" has 1 instance(s).
```

```
  Instance "DBUA0954399", status BLOCKED, has 1 handler(s) for
this service...
Service "cdb1" has 1 instance(s).
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
Service "cdb1XDB" has 1 instance(s).
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
Service "pdb1" has 1 instance(s).
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
Service "pdb2" has 1 instance(s).
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
The command completed successfully

[oracle@host01
```

4. Use the net service name to connect to `pdb2` as `system` user on any of the three instances of the CDB.

```
[oracle@host01 ~]$ sqlplus /nolog
SQL> CONNECT system@"host01:1521/pdb2"
Enter password:
Connected.

SQL> show con_name
CON_NAME
------------------------------
PDB2

SQL> CONNECT system@"host02:1521/pdb2"
Enter password:
Connected.

SQL> CONNECT system@"host03:1521/pdb2"
Enter password:
Connected.

SQL> EXIT
```

Practice 13-3: Affinitizing PDB Services to CDB Instances

Overview

In this practice, you will "affinitize" connections to a PDB to one or particular CDB instances. Because server pools determine which services run together or separately, you can configure and maintain required affinity or isolation.

Tasks

1. Create a dynamic PDB service, `mypdb1serv`, for the `pdb1` PDB in the CDB which will "affinitize" connections to `pdb1` to all the CDB instances.
   a. Check the configuration of the server pools.

```
[oracle@host01 ~]$ srvctl status srvpool
Server pool name: Free
Active servers count: 0
Server pool name: Generic
Active servers count: 0
Server pool name: cdb1pool
Active servers count: 3
[oracle@host01 ~]$

[oracle@host01 ~]$ srvctl config srvpool -serverpool cdb1pool
Server pool name: cdb1pool
Importance: 0, Min: 0, Max: 3
Category: hub
Candidate server names:
[oracle@host01 ~]$
```

   b. Check the services. You notice that the default services created at PDB creation are not managed by the clusterware.

```
[oracle@host01 ~]$ srvctl status service -db cdb1
[oracle@host01 ~]$
[oracle@host01 ~]$ srvctl config service -db cdb1
[oracle@host01 ~]$
```

   c. Create a dynamic PDB service for the `pdb1` PDB in the CDB which will "affinitize" connections to `pdb1` to all the CDB instances uniformly.
      1) Create the service from the connection on the first node of the server pool.

```
[oracle@host01 ~]$ srvctl add service -db cdb1 -pdb pdb1 -
service mypdb1serv -policy automatic -serverpool cdb1pool -
cardinality uniform

[oracle@host01 ~]$ srvctl status service -db cdb1
```

```
Service mypdb1serv is not running.

[oracle@host01 ~]$ srvctl config service -db cdb1
Service name: mypdb1serv
Service is enabled
Server pool: cdb1pool
Cardinality: UNIFORM
Disconnect: false
Service role: PRIMARY
Management policy: AUTOMATIC
DTP transaction: false
AQ HA notifications: false
Global: false
Commit Outcome: false
Failover type:
Failover method:
TAF failover retries:
TAF failover delay:
Connection Load Balancing Goal: LONG
Runtime Load Balancing Goal: NONE
TAF policy specification: NONE
Edition:
Pluggable database name: pdb1
Maximum lag time: ANY
SQL Translation Profile:
Retention: 86400 seconds
Replay Initiation Time: 300 seconds
Session State Consistency:
Service is enabled on nodes:
Service is disabled on nodes:
[oracle@host01 ~]$
```

2) Check that the PDB service is also created on the two other nodes of the server pool.

```
[oracle@host01 ~]$ ssh host02
[oracle@host02 ~]$
[oracle@host02 ~]$ . oraenv
ORACLE_SID = [oracle] ? cdb1_2
ORACLE_HOME = [/home/oracle] ?
/u01/app/oracle/product/12.1.0/dbhome_1
The Oracle base has been set to /u01/app/oracle
```

```
[oracle@host02 ~]$ srvctl status service -db cdb1
Service mypdb1serv is not running.
```

```
[oracle@host02 ~]$ srvctl config service -db cdb1
Service name: mypdb1serv
Service is enabled
Server pool: cdb1pool
Cardinality: UNIFORM
Disconnect: false
Service role: PRIMARY
Management policy: AUTOMATIC
DTP transaction: false
AQ HA notifications: false
Global: false
Commit Outcome: false
Failover type:
Failover method:
TAF failover retries:
TAF failover delay:
Connection Load Balancing Goal: LONG
Runtime Load Balancing Goal: NONE
TAF policy specification: NONE
Edition:
Pluggable database name: pdb1
Maximum lag time: ANY
SQL Translation Profile:
Retention: 86400 seconds
Replay Initiation Time: 300 seconds
Session State Consistency:
Service is enabled on nodes:
Service is disabled on nodes:
[oracle@host02 ~]$ exit
logout
Connection to host02 closed.
[oracle@host02 ~]$
```

You can reiterate the same verification on the third node of the server pool.

```
[oracle@host02 ~]$ ssh host03
[oracle@host03 ~]$ . oraenv
ORACLE_SID = [oracle] ? cdb1_1
ORACLE_HOME = [/home/oracle] ?
/u01/app/oracle/product/12.1.0/dbhome_1
```

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

[oracle@host03 ~]$ srvctl status service -db cdb1
Service mypdb1serv is not running.

[oracle@host03 ~]$ srvctl config service -db cdb1
Service name: mypdb1serv
Service is enabled
Server pool: cdb1pool
Cardinality: UNIFORM
Disconnect: false
Service role: PRIMARY
Management policy: AUTOMATIC
DTP transaction: false
AQ HA notifications: false
Global: false
Commit Outcome: false
Failover type:
Failover method:
TAF failover retries:
TAF failover delay:
Connection Load Balancing Goal: LONG
Runtime Load Balancing Goal: NONE
TAF policy specification: NONE
Edition:
Pluggable database name: pdb1
Maximum lag time: ANY
SQL Translation Profile:
Retention: 86400 seconds
Replay Initiation Time: 300 seconds
Session State Consistency:
Service is enabled on nodes:
Service is disabled on nodes:
[oracle@host03 ~]$ exit
Logout
Connection to host03 closed.

[oracle@host02 ~]$ exit
Logout
Connection to host02 closed.
[oracle@host01 ~]$
```

When services are created with SRVCTL, the `tnsnames.ora` file is not updated and the services are not started.

d. Close the PDB. You will verify that restarting the CDB automatically starts the dynamic PDB service and opens the associated PDB.

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

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP, Advanced Analytics and Real
Application Testing options

SQL> ALTER PLUGGABLE DATABASE pdb1 CLOSE IMMEDIATE
INSTANCES=ALL;
Pluggable database altered.

SQL> SELECT name, open_mode FROM v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB1                           MOUNTED
PDB2                           READ WRITE

SQL> EXIT
```

e. Stop and restart the CDB.

```
[oracle@host01 ~]$ srvctl stop database -d cdb1
[oracle@host01 ~]$
[oracle@host01 ~]$ srvctl start database -db cdb1 -eval
Database cdb1 will be started on nodes host03,host02,host01
Service mypdb1serv will be started on nodes host03,host02,host01
[oracle@host01 ~]$
[oracle@host01 ~]$ srvctl start database -db cdb1

[oracle@host01 ~]$ srvctl status database -db cdb1
Instance cdb1_1 is running on node host03
Instance cdb1_2 is running on node host02
Instance cdb1_3 is running on node host01
[oracle@host01 ~]$
```

f. Verify that the new dynamic PDB service is started and the PDB opened automatically.

```
[oracle@host01 ~]$ srvctl status service -db cdb1
Service mypdb1serv is running on nodes: host03,host02,host01.

[oracle@host01 ~]$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

SQL> SELECT name, open_mode FROM v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB1                           READ WRITE
PDB2                           MOUNTED

SQL> SELECT name FROM v$services;

NAME
----------------------------------------------------------------
mypdb1serv
pdb2
pdb1
cdb1XDB
cdb1
SYS$BACKGROUND
SYS$USERS

7 rows selected.

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

Notice that PDBs are automatically opened by clusterware in all the instances in which the service is started. There is therefore no need to create a trigger `AFTER STARTUP ON DATABASE` to open PDBs as it is the case in non-RAC CDBs.

g. You can also stop and restart the service manually.

```
[oracle@host01 ~]$ srvctl predict service -db cdb1 -service
mypdb1serv
Service mypdb1serv will be stopped on nodes host02,host01,host03

[oracle@host01 ~]$ srvctl stop service -d cdb1 -service
mypdb1serv

[oracle@host01 ~]$ srvctl status service -d cdb1 -service
mypdb1serv
Service mypdb1serv is not running.

[oracle@host01 ~]$ srvctl start service -d cdb1 -service
mypdb1serv

[oracle@host01 ~]$ srvctl status service -d cdb1 -service
mypdb1serv
Service mypdb1serv is running on nodes: host03,host02,host01

[oracle@host01 ~]$ lsnrctl status

LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 17-JUL-
2013 06:07:37

Copyright (c) 1991, 2013, Oracle.  All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.1.0
- Production
Start Date                16-JUL-2013 05:39:02
Uptime                    1 days 0 hr. 28 min. 37 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/12.1.0/grid/network/admin/listener.ora
Listener Log File
/u01/app/grid/diag/tnslsnr/host01/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.101)(PORT=1521
)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.239)(PORT=1521
)))
Services Summary...
Service "+APX" has 1 instance(s).
  Instance "+APX1", status READY, has 1 handler(s) for this
service...
Service "+ASM" has 1 instance(s).
  Instance "+ASM1", status READY, has 2 handler(s) for this
service...
Service "DBUA0954399" has 1 instance(s).
  Instance "DBUA0954399", status BLOCKED, has 1 handler(s) for
this service...
Service "cdb1" has 1 instance(s).
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
Service "cdb1XDB" has 1 instance(s).
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
Service "mypdb1serv" has 1 instance(s).
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
Service "pdb1" has 1 instance(s).
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
Service "pdb2" has 1 instance(s).
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
The command completed successfully
[oracle@host01 ~]$
```

h. Use the service to connect to the PDB on any of the CDB instances.

```
[oracle@host01 ~]$ sqlplus /nolog

SQL> CONNECT system@"host01/mypdb1serv"
Enter password:
Connected.

SQL> SELECT name, open_mode FROM v$pdbs;

NAME                           OPEN_MODE
```

```
------------------------------ ----------
PDB1                           READ WRITE

SQL> CONNECT system@"host02/mypdb1serv"
Enter password:
Connected.

SQL> SELECT name FROM v$services;

NAME
----------------------------------------------------------------
mypdb1serv
pdb1

SQL> CONNECT system@"host03/mypdb1serv"
Enter password:
Connected.

SQL> SELECT name FROM v$services;

NAME
----------------------------------------------------------------
mypdb1serv
pdb1

SQL> EXIT
```

2. You can also "affinitize" connections to `pdb2` to a single node by defining the `mypdb2serv` service cardinality to `SINGLETON`.
   a. Create and start the service for `pdb2`.

```
[oracle@host01 ~]$ srvctl add service -db cdb1 -pdb pdb2 -
service singpdb2serv -policy automatic -serverpool cdb1pool -
cardinality singleton

[oracle@host01 ~]$ srvctl start service -d cdb1 -service
singpdb2serv -eval
Service singpdb2serv will be started on node host01

[oracle@host01 ~]$ srvctl start service -d cdb1 -service
singpdb2serv

[oracle@host01 ~]$ srvctl status service -d cdb1 -service
singpdb2serv
Service singpdb2serv is running on nodes: host01
[oracle@host01 ~]$
```

b. Check that you can use the service to connect to `pdb2` only on `host01` and that the PDB is opened in the CDB instance on `host01` only.

```
[oracle@host01 ~]$ sqlplus /nolog

SQL> CONNECT system@"host01/singpdb2serv"
Enter password:
Connected.

SQL> SELECT name, open_mode FROM v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB2                           READ WRITE

SQL> SELECT name FROM v$services;

NAME
----------------------------------------------------------------
singpdb2serv
pdb2

SQL>
```

c. Check that you cannot use the service to connect to `pdb2` only on `host02` nor `host03` and that the PDB is closed in the CDB instances on `host02` and `host03`.

```
SQL> CONNECT system@"host02/singpdb2serv"
Enter password:
ERROR:
ORA-12514: TNS:listener does not currently know of service
requested in connect descriptor

Warning: You are no longer connected to ORACLE.

SQL> CONNECT system@"host03/singpdb2serv"
Enter password:
ERROR:
ORA-12514: TNS:listener does not currently know of service
requested in connect descriptor

Warning: You are no longer connected to ORACLE.

SQL> CONNECT system@"host02/cdb1"
```

```
Enter password:
Connected.

SQL> select name, open_mode from v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB1                           READ WRITE
PDB2                           MOUNTED

SQL> CONNECT system@"host03/cdb1"
Enter password:
Connected.

SQL> select name, open_mode from v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB1                           READ WRITE
PDB2                           MOUNTED

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

Practice 13-4: Dropping a PDB

Overview

In this practice, you will drop a PDB in the CDB and verify that the services and data files are deleted.

Tasks

1. To drop the pdb2, first stop and remove the service from the resources configuration.

```
[oracle@host01 ~]$ srvctl stop service -d cdb1 -service
singpdb2serv

[oracle@host01 ~]$ srvctl remove service -d cdb1 -service
singpdb2serv
[oracle@host01 ~]$
```

2. Drop the pdb2 PDB.

```
[oracle@host01 ~]$ ~]$ sqlplus /nolog

SQL> CONNECT system@"host01/pdb2"
Enter password:
Connected.

SQL> SELECT name FROM v$datafile;

NAME
------------------------------------------------------------------
----------------
+DATA/CDB1/DATAFILE/undotbs2.294.825668383
+DATA/CDB1/C45A345T5F09726D9C25F01AZ04366B8/DATAFILE/system.268.
335670735
+DATA/CDB1/C45A345T5F09726D9C25F01AZ04366B8/DATAFILE/sysaux.273.
335670729
+DATA/CDB1/C45A345T5F09726D9C25F01AZ04366B8/DATAFILE/users.282.3
35671601

SQL> CONNECT / AS SYSDBA
Connected.

SQL> SELECT name FROM v$services;

NAME
----------------------------------------------------------------
mypdb1serv
pdb2
```

```
pdb1
cdb1XDB
cdb1
SYS$BACKGROUND
SYS$USERS

7 rows selected.

SQL> DROP PLUGGABLE DATABASE pdb2 INCLUDING DATAFILES;
*
ERROR at line 1:
ORA-65025: Pluggable database PDB2 is not closed on all
instances.

SQL> ALTER PLUGGABLE DATABASE pdb2 CLOSE INSTANCES=ALL;

Pluggable database altered.

SQL> DROP PLUGGABLE DATABASE pdb2 INCLUDING DATAFILES;

Pluggable database dropped.

SQL>
```

3. Verify that the data files are deleted.

```
SQL> SELECT name FROM v$datafile;

NAME
--------------------------------------------------------------------
-----------------
+DATA/CDB1/DATAFILE/system.285.825666373
+DATA/CDB1/DATAFILE/sysaux.287.825666251
+DATA/CDB1/DATAFILE/undotbs1.283.825666541
+DATA/CDB1/DD7C48AA5A4404A2E04325AAE80A403C/DATAFILE/system.286.
825666707
+DATA/CDB1/DATAFILE/users.271.825666537
+DATA/CDB1/DD7C48AA5A4404A2E04325AAE80A403C/DATAFILE/sysaux.284.
825666703
+DATA/CDB1/DATAFILE/undotbs2.294.825668383
+DATA/CDB1/DATAFILE/undotbs3.279.825668393
+DATA/CDB1/E5F09726D9C25FC4E043660200C075A9/DATAFILE/system.268.
825670735
```

```
+DATA/CDB1/E5F09726D9C25FC4E043660200C075A9/DATAFILE/sysaux.273.
825670729
+DATA/CDB1/E5F09726D9C25FC4E043660200C075A9/DATAFILE/users.282.8
25671601

11 rows selected.

SQL>
```

Note that all files related to `pdb2` are removed. The UNDO datafile is associated to the instance, and not to any PDB.

4. Verify that the services are deleted. Check in `V$SERVICES` view and with LSNRCTL.

```
SQL> SELECT name FROM v$services;

NAME
----------------------------------------------------------------
mypdb1serv
pdb1
cdb1XDB
cdb1
SYS$BACKGROUND
SYS$USERS

6 rows selected.

SQL> EXIT
[oracle@host01 ~]$ lsnrctl status

LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 17-JUL-
2013 06:07:37

Copyright (c) 1991, 2013, Oracle.  All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.1.0
- Production
Start Date                16-JUL-2013 05:39:02
Uptime                    1 days 0 hr. 28 min. 37 sec
Trace Level               off
Security                  ON: Local OS Authentication
```

```
SNMP                          OFF
Listener Parameter File
/u01/app/12.1.0/grid/network/admin/listener.ora
Listener Log File
/u01/app/grid/diag/tnslsnr/host01/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.101)(PORT=1521
)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.0.2.239)(PORT=1521
)))
Services Summary...
Service "+APX" has 1 instance(s).
  Instance "+APX1", status READY, has 1 handler(s) for this
service...
Service "+ASM" has 1 instance(s).
  Instance "+ASM1", status READY, has 2 handler(s) for this
service...
Service "DBUA0954399" has 1 instance(s).
  Instance "DBUA0954399", status BLOCKED, has 1 handler(s) for
this service...
Service "cdb1" has 1 instance(s).
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
Service "cdb1XDB" has 1 instance(s).
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
Service "mypdb1serv" has 1 instance(s).
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
Service "pdb1" has 1 instance(s).
  Instance "cdb1_3", status READY, has 1 handler(s) for this
service...
The command completed successfully
[oracle@host01 ~]$
```

5. Close all terminal windows opened for this practice.