ORACLE
UNIVERSITY

Hardware and Software
Engineered to Work Together

Oracle Database 12*c*: Clusterware Administration

Activity Guide
D81246GC10
Edition 1.0 | September 2014 | D85451

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

ORACLE

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS
The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Author

Jim Womack

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

This book was published using: Oracle Tutor

Table of Contents

Practices for Lesson 1: Introduction to Clusterware

Chapter 1

Practices for Lesson 1: Overview

Practices Overview

There are no practices for this lesson.

Practices for Lesson 2: Oracle Clusterware Architecture

Chapter 2

Practices for Lesson 2: Overview

Practices Overview

In this practice you will be familiarized with the laboratory environment for this course.

Practice 2-1: Laboratory Introduction

Overview

In this practice you will be familiarized with the laboratory environment for this course.

Tasks

Access to your laboratory environment will be through a graphical display running on your classroom workstation or hosted on a remote machine. Your instructor will provide you with instructions to access your practice environment.

The practice environment for this course is hosted on a server running Oracle Virtual Machine (OVM). In turn, OVM hosts numerous virtual machines (VMs). Each VM is a logically separate server that will be used to run Oracle Database 12c software, including Clusterware, Automatic Storage Management (ASM) and Real Application Clusters (RAC).

1. Open a terminal window and become the `root` user. Execute the `xm list` command. You should see output similar to the example displayed below. It shows that your server is hosting six domains. `Domain-0` is the OVM server. The other domains relate to five VMs which you will use to form a cluster in upcoming practices. Exit the `root` account when finished.

```
[vncuser@classroom_pc ~] su -
Password:  <oracle>
[root@classroom_pc ~]# xm list
Name                           ID   Mem VCPUs      State   Time(s)
Domain-0                        0  1024     4     r-----  18019.8
host01                         24  4200     1     -b----      5.3
host02                         27  3200     1     -b----      6.7
host03                         28  3200     1     -b----      6.6
host04                         25  1600     1     -b----      4.5
host05                         26  1600     1     -b----     33.2
[root@classroom_pc ~]# exit
[vncuser@classroom_pc ~]
```

Check the `Mem` column and take notice of the memory allocated for each VM. The first host, `host01` is allocated 4200M. Next, `host02` and `host03` are allocated 3200M. In addition to Clusterware, they will also host a RAC instance. Lastly, `host04` and `host05` are allocated 1600M. They will become leaf nodes and will not host any database instances.

2. Using SSH, connect to `host01` as the `root` OS user. Enter `oracle` when you are prompted for the password.

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

3. Your environment has been pre-configured with a series of shared disk devices that will be used by Oracle Clusterware and ASM. Execute the following command to identify the shared disks.

```
[root@host01 ~]# ls -l /dev/asmdisk*
brw-rw---- 1 grid asmadmin 202, 81 May  3 16:08 /dev/asmdisk1p1
brw-rw---- 1 grid asmadmin 202, 91 May  3 16:08 /dev/asmdisk1p10
brw-rw---- 1 grid asmadmin 202, 92 May  3 16:08 /dev/asmdisk1p11
brw-rw---- 1 grid asmadmin 202, 93 May  3 16:08 /dev/asmdisk1p12
brw-rw---- 1 grid asmadmin 202, 82 May  3 16:08 /dev/asmdisk1p2
brw-rw---- 1 grid asmadmin 202, 83 May  3 16:08 /dev/asmdisk1p3
brw-rw---- 1 grid asmadmin 202, 85 May  3 16:08 /dev/asmdisk1p4
brw-rw---- 1 grid asmadmin 202, 86 May  3 16:08 /dev/asmdisk1p5
brw-rw---- 1 grid asmadmin 202, 87 May  3 16:08 /dev/asmdisk1p6
brw-rw---- 1 grid asmadmin 202, 88 May  3 16:08 /dev/asmdisk1p7
brw-rw---- 1 grid asmadmin 202, 89 May  3 16:08 /dev/asmdisk1p8
brw-rw---- 1 grid asmadmin 202, 90 May  3 16:08 /dev/asmdisk1p9
brw-rw---- 1 grid asmadmin 202, 97 May  3 16:08 /dev/asmdisk2p1
brw-rw---- 1 grid asmadmin 202, 98 May  3 16:08 /dev/asmdisk2p2
brw-rw---- 1 grid asmadmin 202, 99 May  3 16:08 /dev/asmdisk2p3
brw-rw---- 1 grid asmadmin 202,100 May  3 16:08 /dev/asmdisk2p4
[root@host01 ~]#
```

4. Examine `host02` to confirm that the shared disk devices are also visible on that node.

```
[root@host01 ~]# ssh host02  “ls -l /dev/asmdisk*”
brw-rw---- 1 grid asmadmin 202, 81 May  3 16:08 /dev/asmdisk1p1
brw-rw---- 1 grid asmadmin 202, 91 May  3 16:08 /dev/asmdisk1p10
brw-rw---- 1 grid asmadmin 202, 92 May  3 16:08 /dev/asmdisk1p11
brw-rw---- 1 grid asmadmin 202, 93 May  3 16:08 /dev/asmdisk1p12
brw-rw---- 1 grid asmadmin 202, 82 May  3 16:08 /dev/asmdisk1p2
brw-rw---- 1 grid asmadmin 202, 83 May  3 16:08 /dev/asmdisk1p3
brw-rw---- 1 grid asmadmin 202, 85 May  3 16:08 /dev/asmdisk1p4
brw-rw---- 1 grid asmadmin 202, 86 May  3 16:08 /dev/asmdisk1p5
brw-rw---- 1 grid asmadmin 202, 87 May  3 16:08 /dev/asmdisk1p6
brw-rw---- 1 grid asmadmin 202, 88 May  3 16:08 /dev/asmdisk1p7
brw-rw---- 1 grid asmadmin 202, 89 May  3 16:08 /dev/asmdisk1p8
brw-rw---- 1 grid asmadmin 202, 90 May  3 16:08 /dev/asmdisk1p9
brw-rw---- 1 grid asmadmin 202, 97 May  3 16:08 /dev/asmdisk2p1
brw-rw---- 1 grid asmadmin 202, 98 May  3 16:08 /dev/asmdisk2p2
brw-rw---- 1 grid asmadmin 202, 99 May  3 16:08 /dev/asmdisk2p3
brw-rw---- 1 grid asmadmin 202,100 May  3 16:08 /dev/asmdisk2p4
[root@host01 ~]#
```

5. Examine `host03` to confirm that the shared disk devices are also visible on that node.

```
[root@host01 ~]# ssh host03  “ls -l /dev/asmdisk*”
brw-rw---- 1 grid asmadmin 202, 81 May  3 16:08 /dev/asmdisk1p1
brw-rw---- 1 grid asmadmin 202, 91 May  3 16:08 /dev/asmdisk1p10
brw-rw---- 1 grid asmadmin 202, 92 May  3 16:08 /dev/asmdisk1p11
brw-rw---- 1 grid asmadmin 202, 93 May  3 16:08 /dev/asmdisk1p12
brw-rw---- 1 grid asmadmin 202, 82 May  3 16:08 /dev/asmdisk1p2
brw-rw---- 1 grid asmadmin 202, 83 May  3 16:08 /dev/asmdisk1p3
brw-rw---- 1 grid asmadmin 202, 85 May  3 16:08 /dev/asmdisk1p4
brw-rw---- 1 grid asmadmin 202, 86 May  3 16:08 /dev/asmdisk1p5
brw-rw---- 1 grid asmadmin 202, 87 May  3 16:08 /dev/asmdisk1p6
brw-rw---- 1 grid asmadmin 202, 88 May  3 16:08 /dev/asmdisk1p7
brw-rw---- 1 grid asmadmin 202, 89 May  3 16:08 /dev/asmdisk1p8
brw-rw---- 1 grid asmadmin 202, 90 May  3 16:08 /dev/asmdisk1p9
brw-rw---- 1 grid asmadmin 202, 97 May  3 16:08 /dev/asmdisk2p1
brw-rw---- 1 grid asmadmin 202, 98 May  3 16:08 /dev/asmdisk2p2
brw-rw---- 1 grid asmadmin 202, 99 May  3 16:08 /dev/asmdisk2p3
brw-rw---- 1 grid asmadmin 202,100 May  3 16:08 /dev/asmdisk2p4
[root@host01 ~]#
```

6. Examine `host04` and `host05` as well. Notice that the devices are not available on these nodes. You will see how these nodes can participate in the cluster in upcoming practices.

```
[root@host01 ~]# ssh host04 “ls -l /dev/asmdisk*”
ls: cannot access /dev/asmdisk*: No such file or directory

[root@host01 ~]# ssh host05 “ls -l /dev/asmdisk*”
ls: cannot access /dev/asmdisk*: No such file or directory
[root@host01 ~]#
```

So far you have examined the cluster nodes and identified the two main differences (the amount of system memory and the availability of shared storage) between them.

7. The nodes have all been preconfigured with the requirements for installing Oracle Database 12c. This includes the OS user accounts required for a role separated installation. Using `su`, assume the identity of the `grid` user.

```
[root@host01 ~]# su - grid
[grid@host01 ~]$
```

8. Examine the `grid` user account and take note of the OS groups that it is associated with.

```
[grid@host01 ~]$ id
uid=54322(grid) gid=54321(oinstall)
groups=54321(oinstall),54327(asmdba),54328(asmoper),54329(asmadm
in)
[grid@host01 ~]$
```

9. Exit the `grid` session.

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

10. Using `su`, assume the identity of the `oracle` user and examine the user account to identify the OS groups associated with it.

```
[root@host01 ~]# su - oracle
[oracle@host01 ~]$ id
uid=54321(oracle) gid=54321(oinstall)
groups=54321(oinstall),54322(dba),54323(oper),54327(asmdba)
[oracle@host01 ~]$
```

The `root`, `grid` and `oracle` OS user accounts have all been pre-configured to enable passwordless SSH between the cluster nodes. For the `grid` and `oracle` users, this is required to install Oracle Database 12c. For the `root` user, this configuration simplifies some of the practice tasks (as you have already seen in steps 4 and 5 above).

11. Confirm the configuration of passwordless SSH for the `oracle` and `grid` OS users.

```
[oracle@host01 bin]$ ssh host02
[oracle@host02 ~]$ logout
Connection to host02 closed.
[oracle@host01 bin]$ ssh host03
[oracle@host03 ~]$ logout
Connection to host03 closed.
[oracle@host01 bin]$ ssh host04
[oracle@host04 ~]$ logout
Connection to host04 closed.
[oracle@host01 bin]$ ssh host05
[oracle@host05 ~]$ logout
Connection to host05 closed.
[oracle@host01 bin]$ logout

[root@host01 ~]# su - grid
[grid@host01 ~]$ ssh host02
[grid@host02 ~]$ exit
```

```
logout
Connection to host02 closed.
[grid@host01 ~]$ ssh host03
[grid@host03 ~]$ logout
Connection to host03 closed.
[grid@host01 ~]$ ssh host04
[grid@host04 ~]$ logout
Connection to host04 closed.
[grid@host01 ~]$ ssh host05
[grid@host05 ~]$ logout

Connection to host05 closed.
[grid@host01 ~]$
```

12. Exit your terminal session.

```
[grid@host01 ~]$ logout

[root@host01 ~]# logout

[root@host01 ~]# logout

Connection to host01 closed.
[vncuser@classroom_pc ~] $
```

Note: Each of the following practice exercises will instruct you to initiate one or more terminal sessions using different user accounts. Ensure that you always start a new terminal session when instructed to do so, and exit all of your terminal sessions at the end of each practice.

Practices for Lesson 3: Flex Clusters

Chapter 3

Practices for Lesson 3: Overview

Practices Overview

There are no practices for this lesson.

Practices for Lesson 4: Grid Infrastructure Preinstallation Tasks

Chapter 4

Practices for Lesson 4: Overview

Practices Overview

In this practice, you perform various tasks that are required before installing Oracle Grid Infrastructure.

Practice 4-1: Preinstallation Tasks

Overview

In this practice you will perform required preinstallation tasks for Oracle12c Grid Infrastructure.

Tasks

You will perform various tasks that are required before installing Oracle Grid Infrastructure. These tasks include:

- Creating base directory
- Configuring Network Time Protocol (NTPD)
- Setting shell limits
- Editing profile entries

1. SSH to `host01` as the root user. View the `/etc/sysconfig/ntpd` file and confirm that the `-x` option is specified to address slewing. If necessary, change the file, and then restart the ntpd service with the `service ntpd restart` command. **Perform this step on all five of your nodes**.

```
[vncuser@classroom_pc ~] $ ssh root@host01
Password:  <oracle>
[root@host01 ~]# cat /etc/sysconfig/ntpd
# Drop root to id 'ntp:ntp' by default.
OPTIONS="-x -u ntp:ntp -p /var/run/ntpd.pid"

# Set to 'yes' to sync hw clock after successful ntpdate
SYNC_HWCLOCK=no

# Additional options for ntpdate
NTPDATE_OPTIONS=""
#
```

2. As the `root` user, start the local naming cache daemon on all five 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 five of your nodes**.

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

[root@host01 ~]# ssh host02 service nscd start
root's password: 0racle << password is not displayed
Starting nscd: [  OK  ]
[root@host01 ~]# ssh host02 chkconfig nscd on
root's password: 0racle << password is not displayed

[root@host01 ~]# ssh host03 service nscd start
root's password: 0racle << password is not displayed
Starting nscd: [  OK  ]
```

```
[root@host01 ~]# ssh host03 chkconfig nscd on
root's password: 0racle << password is not displayed

[root@host01 ~]# ssh host04 service nscd start
root's password: oracle << password is not displayed
Starting nscd: [  OK  ]
[root@host01 ~]# ssh host04 chkconfig nscd on
root's password: oracle << password is not displayed

[root@host01 ~]# ssh host05 service nscd start
root's password: oracle << password is not displayed
Starting nscd: [  OK  ]
[root@host01 ~]# ssh host05 chkconfig nscd on
root's password: oracle << password is not displayed
```

3. As the `root` user, run the `/stage/GRID/labs/less_04/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`. Cat `the /stage/GRID/labs/less_04/bash_profile` and `/stage/GRID/labs/less_04/profile` to view the new files. It also installs the CVU rpm. **Perform this step on all five of your node.**

```
[root@host01 ~]# cat /stage/GRID/labs/less_04/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/GRID/labs/less_04/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/GRID/labs/less_04/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 ~]# cat /stage/GRID/labs/less_04/limits.sh
cp /stage/GRID/labs/less_04/bash_profile /home/oracle/.bash_profile
cp /stage/GRID/labs/less_04/bash_profile /home/grid/.bash_profile
cp /stage/GRID/labs/less_04/limits.conf /etc/security/limits.conf
cp /etc/profile /root/etc_profile
cp /stage/GRID/labs/less_04/profile /etc/profile


ssh host02 cp /stage/GRID/labs/less_04/bash_profile
/home/oracle/.bash_profile
ssh host02 cp /stage/GRID/labs/less_04/bash_profile
/home/grid/.bash_profile
```

```
ssh host02 cp /stage/GRID/labs/less_04/limits.conf
/etc/security/limits.conf
ssh host02 cp /etc/profile /root/etc_profile
ssh host02 cp /stage/GRID/labs/less_04/profile /etc/profile


ssh host03 cp /stage/GRID/labs/less_04/bash_profile
/home/oracle/.bash_profile
ssh host03 cp /stage/GRID/labs/less_04/bash_profile
/home/grid/.bash_profile
ssh host03 cp /stage/GRID/labs/less_04/limits.conf
/etc/security/limits.conf
ssh host03 cp /etc/profile /root/etc_profile
ssh host03 cp /stage/GRID/labs/less_04/profile /etc/profile

ssh host04 cp /stage/GRID/labs/less_04/bash_profile
/home/oracle/.bash_profile
ssh host04 cp /stage/GRID/labs/less_04/bash_profile
/home/grid/.bash_profile
ssh host04 cp /stage/GRID/labs/less_04/limits.conf
/etc/security/limits.conf
ssh host04 cp /etc/profile /root/etc_profile
ssh host04 cp /stage/GRID/labs/less_04/profile /etc/profile

ssh host05 cp /stage/GRID/labs/less_04/bash_profile
/home/oracle/.bash_profile
ssh host05 cp /stage/GRID/labs/less_04/bash_profile
/home/grid/.bash_profile
ssh host05 cp /stage/GRID/labs/less_04/limits.conf
/etc/security/limits.conf
ssh host05 cp /etc/profile /root/etc_profile
ssh host05 cp /stage/GRID/labs/less_04/profile /etc/profile

rpm -iv /stage/clusterware/rpm/cvuqdisk-1.0.9-1.rpm
ssh host02 rpm -iv /stage/clusterware/rpm/cvuqdisk-1.0.9-1.rpm
ssh host03 rpm -iv /stage/clusterware/rpm/cvuqdisk-1.0.9-1.rpm
ssh host04 rpm -iv /stage/clusterware/rpm/cvuqdisk-1.0.9-1.rpm
ssh host05 rpm -iv /stage/clusterware/rpm/cvuqdisk-1.0.9-1.rpm

[root@host01 ~]# /stage/GRID/labs/less_04/limits.sh
...
```

4. Make sure the Oracle Pre-Install RPM is installed on all five of your cluster nodes.

```
[root@host01 ~]# rpm -qa|grep -i preinstall
oracle-rdbms-server-12cR1-preinstall-1.0-8.el6.x86_64

[root@host01 ~]# ssh host02 rpm -qa|grep -i preinstall
oracle-rdbms-server-12cR1-preinstall-1.0-8.el6.x86_64

[root@host01 ~]# ssh host03 rpm -qa|grep -i preinstall
```

```
oracle-rdbms-server-12cR1-preinstall-1.0-8.el6.x86_64

[root@host01 ~]# ssh host04 rpm -qa|grep -i preinstall
oracle-rdbms-server-12cR1-preinstall-1.0-8.el6.x86_64

[root@host01 ~]# ssh host05 rpm -qa|grep -i preinstall
oracle-rdbms-server-12cR1-preinstall-1.0-8.el6.x86_64

[root@host01 ~]#
```

5. Check that the proper scheduler is used for your shared disks (actual devices are `xvde` and `xvdf`). On each cluster node, enter the following commands to ensure that the Deadline disk I/O scheduler is configured for use:

```
[root@host01 ~]# cat /sys/block/xvde/queue/scheduler
noop [deadline] cfq

[root@host01 ~]# cat /sys/block/xvdf/queue/scheduler
noop [deadline] cfq

[root@host01 ~]# ssh host02 cat /sys/block/xvde/queue/scheduler
noop [deadline] cfq

[root@host01 ~]# ssh host02 cat /sys/block/xvdf/queue/scheduler
noop [deadline] cfq

[root@host01 ~]# ssh host03 cat /sys/block/xvde/queue/scheduler
noop [deadline] cfq

[root@host01 ~]# ssh host03cat /sys/block/xvdf/queue/scheduler
noop [deadline] cfq

[root@host01 ~]#
```

6. Create the installation directories for both `grid` and `oracle`-owned software. Set the ownership and permissions of the directories as shown below: Do this on all five hosts. Use the `/stage/GRID/labs/less_04/cr_dir.sh` script to save time.

```
[root@host01 ~]# cat /stage/GRID/labs/less_04/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 host04 mkdir -p /u01/app/12.1.0/grid
ssh host04 mkdir -p /u01/app/grid
ssh host04 mkdir -p /u01/app/oracle
ssh host04 chown -R grid:oinstall /u01
ssh host04 chown oracle:oinstall /u01/app/oracle
ssh host04 chmod -R 775 /u01/

ssh host05 mkdir -p /u01/app/12.1.0/grid
ssh host05 mkdir -p /u01/app/grid
ssh host05 mkdir -p /u01/app/oracle
ssh host05 chown -R grid:oinstall /u01
ssh host05 chown oracle:oinstall /u01/app/oracle
ssh host05 chmod -R 775 /u01/

[root@host01 ~]# /stage/GRID/labs/less_04/cr_dir.sh
...
[root@host01 ~]#
```

7. Close all terminal windows opened for these practices.

Practices for Lesson 5: Grid Infrastructure Installation

Chapter 5

Practices for Lesson 5: Overview

Practices Overview

In these practices, you will:

- Configure a new Flex Cluster with Flex ASM.

Practice 5-1: Configuring a new Flex Cluster with Flex ASM

Overview

In this practice you will install and configure a new Flex Cluster with Flex ASM. You will install to three nodes; `host01`, `host02` and `host04`. You will designate `host01` and `host02` to be hub nodes and `host04` will be designated as a leaf node.

1. From a `vncuser` terminal on your desktop PC, change to the `root` account. First, 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 -
Password:

[root@classroom_pc ~]# TIME="`date +%T`";for H in host01 host02
host03 host04 host05;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:                                           [  OK  ]
[root@classroom_pc ~]# exit
Logout
[vncuser@classroom_pc - ~]$
```

2. 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.

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

3. 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, then restart the installer ####
```

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

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

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

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

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

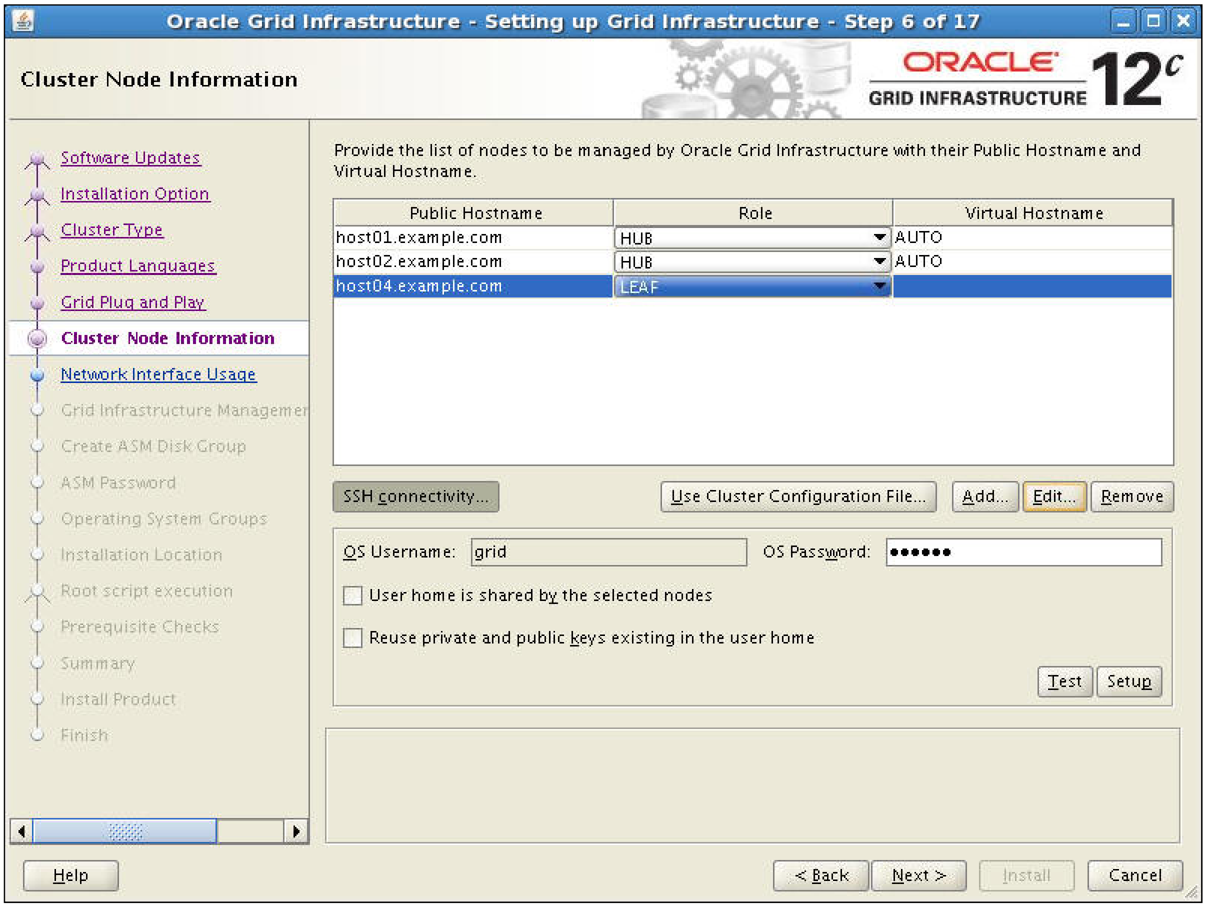
8. Use the Grid Plug and Play Information screen to configure the following settings:
   - Cluster Name: `cluster01`
   - SCAN Name: `cluster01-scan.cluster01.example.com`
   - SCAN Port: `1521`
   - GNS VIP Address: `192.0.2.155`
   - GNS Sub Domain: `cluster01.example.com`

   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.

9. 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 `host04.example.com`. Make sure the Node Role is set to **`LEAF`** and click OK.

10. 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. Your screen Cluster Node Information page should look like the one below:



Provide the list of nodes to be managed by Oracle Grid Infrastructure with their Public Hostname and Virtual Hostname.

Public Hostname	Role	Virtual Hostname
host01.example.com	HUB	AUTO
host02.example.com	HUB	AUTO
host04.example.com	LEAF	

OS Username: grid   OS Password: ●●●●●●

□ User home is shared by the selected nodes

□ Reuse private and public keys existing in the user home

11. 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 “Do not use”. Click Next to continue.

12. On the Grid Infrastructure Management Repository screen, select No and click Next to continue. Click Yes in the dialog box to confirm your choice.

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

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

15. 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.

16. 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.

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

18. 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.

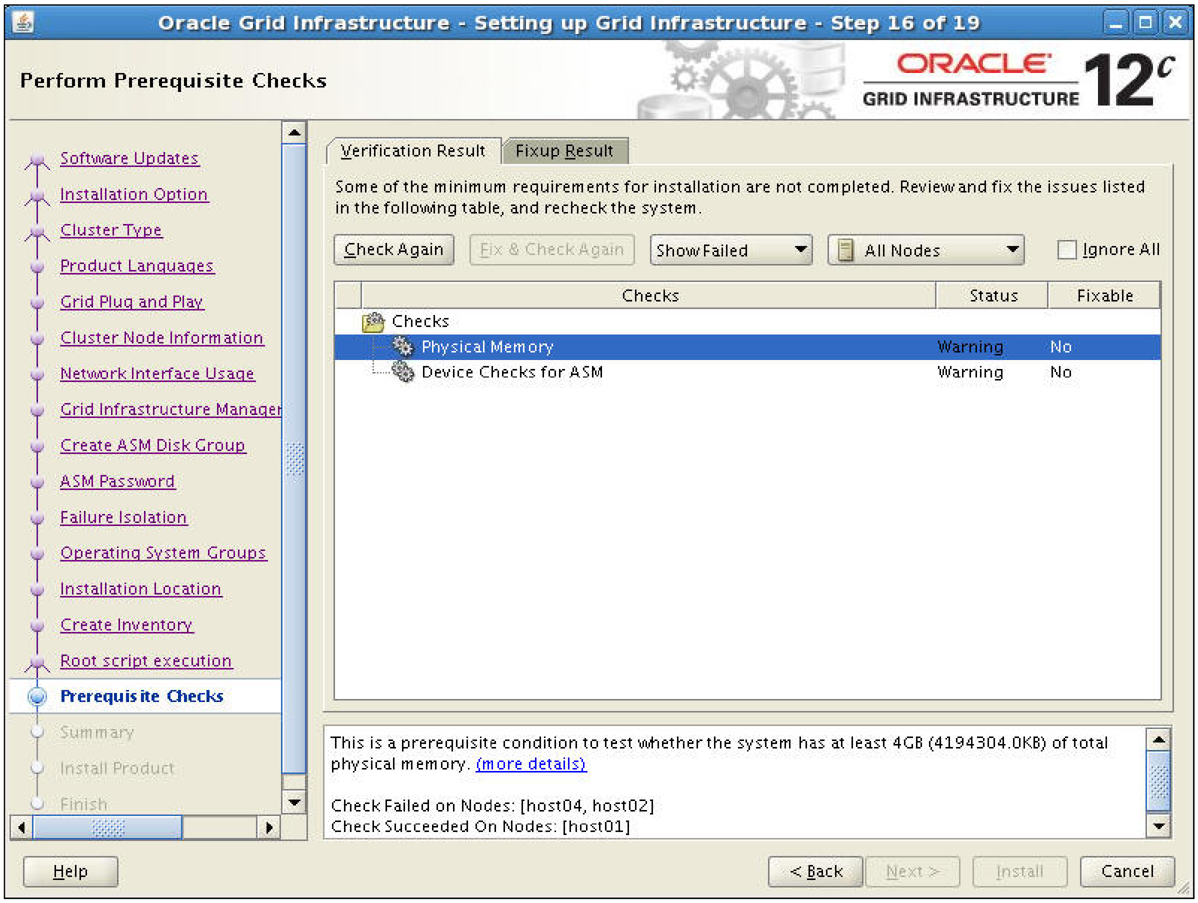
19. On the Specify Installation Location screen, make sure the Oracle base is **`/u01/app/grid`** and change the Software location to **`/u01/app/12.1.0/grid`**. Click Next to continue.

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

21. 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.

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

23. Due to 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.



Oracle Grid Infrastructure - Setting up Grid Infrastructure - Step 16 of 19

Perform Prerequisite Checks

Verification Result | Fixup Result

Some of the minimum requirements for installation are not completed. Review and fix the issues listed in the following table, and recheck the system.

Checks	Status	Fixable
Checks		
Physical Memory	Warning	No
Device Checks for ASM	Warning	No

This is a prerequisite condition to test whether the system has at least 4GB (4194304.0KB) of total physical memory. (more details)

Check Failed on Nodes: [host04, host02]
Check Succeeded On Nodes: [host01]

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

25. 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.

26. 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.

27. After configuration completes you will see the following message:

    "The installation of Oracle Grid Infrastucture for a Cluster was successful"

    Click Close to close Oracle Universal Installer.

28. 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 ~]$
```

29. 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
**************************************************************
host04:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
[grid@host01 ~]$
```

30. 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.ASMNET1LSNR_ASM.lsnr`
    - Leaf listeners: `ora.LISTENER_LEAF.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
ora.DATA.dg
               ONLINE  ONLINE       host01                STABLE
               ONLINE  ONLINE       host02                STABLE
ora.LISTENER.lsnr
               ONLINE  ONLINE       host01                STABLE
               ONLINE  ONLINE       host02                STABLE
```

```
ora.LISTENER_LEAF.lsnr
               OFFLINE OFFLINE      host04                   STABLE
ora.net1.network
               ONLINE  ONLINE       host01                   STABLE
               ONLINE  ONLINE       host02                   STABLE
ora.ons
               ONLINE  ONLINE       host01                   STABLE
               ONLINE  ONLINE       host02                   STABLE
ora.proxy_advm
               ONLINE  ONLINE       host01                   STABLE
               ONLINE  ONLINE       host02                   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       host01                   STABLE

ora.asm
      1        ONLINE  ONLINE       host01                   STABLE
      2        ONLINE  ONLINE       host02                   STABLE
      3        OFFLINE OFFLINE                               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.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       host01                 STABLE
----------------------------------------------------------------
[grid@host01 ~]$
```

31. Close all terminal windows opened for these practices.

Practices for Lesson 6: Managing Cluster Nodes

Chapter 6

Practices for Lesson 6: Overview

Practices Overview

In these practices, you will:

- Add a new Hub node to your cluster
- Add a new Leaf node to your cluster.
- Install Oracle RAC database software
- Create a RAC database

Practice 6-1: Add a New Hub Node to the Cluster

Overview

In this practice, you will use `addnode.sh` in graphical mode to extend your cluster by adding `host03` as the third hub node.

1. 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.

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

2. Make sure that you can connect from your `host01` to `host03` without being prompted for passwords. Exit back to `host01` when finished.

```
[grid@host01 ~]$ ssh host03
[grid@host03 ~]$ exit
logout
Connection to host03 closed.
[grid@host01 ~]$
```

3. Make sure that you set up your environment variables correctly for the `grid` user to point to your grid installation.

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

4. Check your pre-grid installation for `host03` node using the Cluster Verification Utility.

```
[grid@host01 ~]$ cluvfy stage -pre crsinst -n host03

Performing pre-checks for cluster services setup

Checking node reachability...
Node reachability check passed from node "host01"



Checking user equivalence...
User equivalence check passed for user "grid"

Checking node connectivity...

Checking hosts config file...

Verification of the hosts config file successful

Check: Node connectivity using interfaces on subnet "192.0.2.0"
Node connectivity passed for subnet "192.0.2.0" with node(s)
host03
TCP connectivity check passed for subnet "192.0.2.0"


Check: Node connectivity using interfaces on subnet
"192.168.1.0"
Node connectivity passed for subnet "192.168.1.0" with node(s)
host03
TCP connectivity check passed for subnet "192.168.1.0"


Node connectivity check passed

Checking multicast communication...

Checking subnet "192.168.1.0" for multicast communication with
multicast group "224.0.0.251"...
Check of subnet "192.168.1.0" for multicast communication with
multicast group "224.0.0.251" passed.

Check of multicast communication passed.
Total memory check failed
Check failed on nodes:
        host03
Available memory check passed
Swap space check passed
Free disk space check passed for
"host03:/usr,host03:/var,host03:/etc,host03:/sbin,host03:/tmp"
Free disk space check passed for "host03:/u01/app/12.1.0/grid"
Check for multiple users with UID value 54322 passed
User existence check passed for "grid"
Group existence check passed for "oinstall"
Group existence check passed for "dba"
Membership check for user "grid" in group "oinstall" [as
Primary] passed
Membership check for user "grid" in group "dba" failed
Check failed on nodes:
        host03
```

```
Run level check passed
Hard limits check passed for "maximum open file descriptors"
Soft limits check passed for "maximum open file descriptors"
Hard limits check passed for "maximum user processes"
Soft limits check passed for "maximum user processes"
System architecture check passed
Kernel version check passed
Kernel parameter check passed for "semmsl"
Kernel parameter check passed for "semmns"
Kernel parameter check passed for "semopm"
Kernel parameter check passed for "semmni"
Kernel parameter check passed for "shmmax"
Kernel parameter check passed for "shmmni"
Kernel parameter check passed for "shmall"
Kernel parameter check passed for "file-max"
Kernel parameter check passed for "ip_local_port_range"
Kernel parameter check passed for "rmem_default"
Kernel parameter check passed for "rmem_max"
Kernel parameter check passed for "wmem_default"
Kernel parameter check passed for "wmem_max"
Kernel parameter check passed for "aio-max-nr"
Package existence check passed for "binutils"
Package existence check passed for "compat-libcap1"
Package existence check passed for "compat-libstdc++-33(x86_64)"
Package existence check passed for "libgcc(x86_64)"
Package existence check passed for "libstdc++(x86_64)"
Package existence check passed for "libstdc++-devel(x86_64)"
Package existence check passed for "sysstat"
Package existence check passed for "gcc"
Package existence check passed for "gcc-c++"
Package existence check passed for "ksh"
Package existence check passed for "make"
Package existence check passed for "glibc(x86_64)"
Package existence check passed for "glibc-devel(x86_64)"
Package existence check passed for "libaio(x86_64)"
Package existence check passed for "libaio-devel(x86_64)"
Package existence check passed for "nfs-utils"
Check for multiple users with UID value 0 passed
Current group ID check passed

Starting check for consistency of primary group of root user
```

```
Check for consistency of root user's primary group passed

Starting Clock synchronization checks using Network Time
Protocol(NTP)...

NTP Configuration file check started...
NTP Configuration file check passed

Checking daemon liveness...
Liveness check passed for "ntpd"
Check for NTP daemon or service alive passed on all nodes

NTP common Time Server Check started...
Check of common NTP Time Server passed

Clock time offset check from NTP Time Server started...
Clock time offset check passed

Clock synchronization check using Network Time Protocol(NTP)
passed

Core file name pattern consistency check passed.

User "grid" is not part of "root" group. Check passed
Default user file creation mask check passed
Time zone consistency check passed

Checking integrity of name service switch configuration file
"/etc/nsswitch.conf" ...
Check for integrity of name service switch configuration file
"/etc/nsswitch.conf" passed


Checking daemon "avahi-daemon" is not configured and running
Daemon not configured check passed for process "avahi-daemon"
Daemon not running check passed for process "avahi-daemon"

Starting check for Reverse path filter setting ...

Check for Reverse path filter setting passed

Starting check for /dev/shm mounted as temporary file system ...
```

```
Check for /dev/shm mounted as temporary file system passed

Pre-check for cluster services setup was unsuccessful on all the
nodes.
[grid@host01 Disk1]$
[grid@host01 ~]$
```

The only exceptions you should encounter are total system memory and grid membership in the dba group, which is normal.

5. Change directory to `$ORACLE_HOME/addnode` and execute `addnode.sh`.

```
[grid@host01 ~]$ cd $ORACLE_HOME/addnode
[grid@host01 addnode]$ ./addnode.sh
```

6. On the Cluster Add Node Information screen, click Add to add the hostname for the node to be added.

7. Enter `host03.example.com` for the Public Hostname and select HUB as the Node Role. Click OK, then click Next to continue.

8. On the Perform Prerequisite Checks page, the Physical memory test will fail as host03 has less than 4GB of memory. Click the Ignore All check box and click Next.

9. Click Yes in the Oracle Grid Infrastructure dialog box.

10. Click Install on the Summary page.

11. The Execute Configuration Scripts dialog box will prompt you to run the `root` scripts on `host03`. Run the `orainstRoot.sh` script first.

```
[root@host01 ~]# ssh host03 /u01/app/oraInventory/orainstRoot.sh
Changing permissions of /u01/app/oraInventory.
Adding read,write permissions for group.
Removing read,write,execute permissions for world.

Changing groupname of /u01/app/oraInventory to oinstall.
The execution of the script is complete.
```

12. Next, execute the `root.sh`.script on `host03`.

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

```
The following environment variables are set as:
    ORACLE_OWNER= grid
    ORACLE_HOME=  /u01/app/12.1.0/grid

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.



Creating /etc/oratab file...
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.
Relinking oracle with rac_on option
Using configuration parameter file:
/u01/app/12.1.0/grid/crs/install/crsconfig_params
2013/05/16 11:14:28 CLSRSC-363: User ignored prerequisites
during installation

OLR initialization - successful
2013/05/16 11:14:56 CLSRSC-330: Adding Clusterware entries to
file 'oracle-ohasd.conf'

CRS-4133: Oracle High Availability Services has been stopped.
CRS-4123: Oracle High Availability Services has been started.
CRS-4133: Oracle High Availability Services has been stopped.
CRS-4123: Oracle High Availability Services has been started.
CRS-2791: Starting shutdown of Oracle High Availability
Services-managed resources on 'host03'
CRS-2673: Attempting to stop 'ora.drivers.acfs' on 'host03'
CRS-2677: Stop of 'ora.drivers.acfs' on 'host03' succeeded
CRS-2793: Shutdown of Oracle High Availability Services-managed
resources on 'host03' has completed
CRS-4133: Oracle High Availability Services has been stopped.
CRS-4123: Starting Oracle High Availability Services-managed
resources
CRS-2672: Attempting to start 'ora.mdnsd' on 'host03'
CRS-2672: Attempting to start 'ora.evmd' on 'host03'
CRS-2676: Start of 'ora.mdnsd' on 'host03' succeeded
CRS-2676: Start of 'ora.evmd' on 'host03' succeeded
```

```
CRS-2672: Attempting to start 'ora.gpnpd' on 'host03'
CRS-2676: Start of 'ora.gpnpd' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.gipcd' on 'host03'
CRS-2676: Start of 'ora.gipcd' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'host03'
CRS-2676: Start of 'ora.cssdmonitor' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.cssd' on 'host03'
CRS-2672: Attempting to start 'ora.diskmon' on 'host03'
CRS-2676: Start of 'ora.diskmon' on 'host03' succeeded
CRS-2789: Cannot stop resource 'ora.diskmon' as it is not
running on server 'host03'
CRS-2676: Start of 'ora.cssd' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.cluster_interconnect.haip' on
'host03'
CRS-2672: Attempting to start 'ora.ctssd' on 'host03'
CRS-2676: Start of 'ora.ctssd' on 'host03' succeeded
CRS-2676: Start of 'ora.cluster_interconnect.haip' on 'host03'
succeeded
CRS-2672: Attempting to start 'ora.asm' on 'host03'
CRS-2676: Start of 'ora.asm' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.storage' on 'host03'
CRS-2676: Start of 'ora.storage' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.crf' on 'host03'
CRS-2676: Start of 'ora.crf' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'host03'
CRS-2676: Start of 'ora.crsd' on 'host03' succeeded
CRS-6017: Processing resource auto-start for servers: host03
CRS-2673: Attempting to stop 'ora.LISTENER_SCAN2.lsnr' on
'host01'
CRS-2672: Attempting to start 'ora.ASMNET1LSNR_ASM.lsnr' on
'host03'
CRS-2672: Attempting to start 'ora.ons' on 'host03'
CRS-2677: Stop of 'ora.LISTENER_SCAN2.lsnr' on 'host01'
succeeded
CRS-2673: Attempting to stop 'ora.scan2.vip' on 'host01'
CRS-2677: Stop of 'ora.scan2.vip' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.scan2.vip' on 'host03'
CRS-2676: Start of 'ora.scan2.vip' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.LISTENER_SCAN2.lsnr' on
'host03'
CRS-2676: Start of 'ora.ASMNET1LSNR_ASM.lsnr' on 'host03'
succeeded
CRS-2672: Attempting to start 'ora.asm' on 'host03'
CRS-2676: Start of 'ora.ons' on 'host03' succeeded
```

```
CRS-2676: Start of 'ora.LISTENER_SCAN2.lsnr' on 'host03'
succeeded
CRS-2676: Start of 'ora.asm' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.host03.vip' on 'host03'
CRS-2676: Start of 'ora.host03.vip' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.LISTENER.lsnr' on 'host03'
CRS-2676: Start of 'ora.LISTENER.lsnr' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.proxy_advm' on 'host03'
CRS-2676: Start of 'ora.proxy_advm' on 'host03' succeeded
CRS-6016: Resource auto-start has completed for server host03
CRS-6024: Completed start of Oracle Cluster Ready Services-
managed resources
CRS-4123: Oracle High Availability Services has been started.
2013/05/16 11:20:06 CLSRSC-343: Successfully started Oracle
clusterware stack

clscfg: EXISTING configuration version 5 detected.
clscfg: version 5 is 12c Release 1.
Successfully accumulated necessary OCR keys.
Creating OCR keys for user 'root', privgrp 'root'..
Operation successful.

2013/05/16 11:20:19 CLSRSC-326: Configure Oracle Grid
Infrastructure for a Cluster ... Succeeded
```

Note: If the root.sh script indicates a failure relating to starting the listener on host03: 2014/01/14 05:01:39 CLSRSC-180: An error occurred while executing the command '/u01/app/12.1.0/grid/bin/srvctl start listener -n host03' (error code 512)

Verify that all local and cluster resources are up on host03 by running `crsctl stat res -t`. All resources should indicate online including the listener, and you can continue safely.

13. When the `root` scripts have finished running on host03, return to the Execute Configuration Scripts dialog box and click OK.

14. Click Close on the Finish screen when prompted by the OUI.

15. Set your environment and execute the `crsctl stat res -t` command to verify that all local and cluster resources are running on `host03` as expected.

```
[root@host01 ~]# . oraenv
ORACLE_SID = [root] ? +ASM1
The Oracle base has been set to /u01/app/grid
[root@host01 ~]# crsctl stat res -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.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.LISTENER_LEAF.lsnr
               OFFLINE OFFLINE      host04                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       host03                STABLE
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       host01                STABLE
ora.MGMTLSNR
      1        ONLINE  ONLINE       host01  169.254.78.72 192.16
```

```
                                                  8.1.101,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       host03                STABLE
ora.scan3.vip
      1        ONLINE  ONLINE       host01                STABLE
----------------------------------------------------------------
[root@host01 ~]#
```

Practice 6-2: Add a New Leaf Node to the Cluster

Overview

In this practice, you will use `addnode.sh` in silent mode to extend your cluster by adding `host05` as the second leaf node.

1. Establish a terminal session connected to `host01` as the `grid` OS user. Do ***not*** specify the **`-X`** option for `ssh`.

```
[grid@host01 addnode]$ exit  (From the grid terminal opened
                              in  the previous practice)

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

2. Make sure that you can connect from your `host01` to `host05` without being prompted for passwords. Exit back to `host01` when finished.

```
[grid@host01 ~]$ ssh host05
[grid@host05 ~]$ exit
logout
Connection to host03 closed.
[grid@host01 ~]$
```

3. Execute **`.`** `oraenv` to set your environment. Set the SID to `+ASM1`.

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

4. Change directory to `$ORACLE_HOME/addnode` and run `addnode.sh` silently and add `host05` to your cluster as a leaf node.

```
[grid@host01 ~]$ cd $ORACLE_HOME/addnode
[grid@host01 addnode]$ ./addnode.sh -silent
"CLUSTER_NEW_NODES={host05}" "CLUSTER_NEW_NODE_ROLES={leaf}"
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 120 MB.   Actual 7949
MB    Passed
Checking swap space: must be greater than 150 MB.   Actual 8632
MB    Passed
[WARNING] [INS-13014] Target environment does not meet some
optional requirements.
```

```
   CAUSE: Some of the optional prerequisites are not met. See
logs for details. /u01/app/oraInventory/logs/addNodeActions2013-
05-16_04-07-37PM.log
   ACTION: Identify the list of failed prerequisite checks from
the log: /u01/app/oraInventory/logs/addNodeActions2013-05-16_04-
07-37PM.log. Then either from the log file or from installation
manual find the appropriate configuration to meet the
prerequisites and fix it manually.


Prepare Configuration in progress.


Prepare Configuration successful.
..................................................   9% Done.
You can find the log of this install session at:
 /u01/app/oraInventory/logs/addNodeActions2013-05-16_04-07-
37PM.log


Instantiate files in progress.


Instantiate files successful.
..................................................   15% Done.


Copying files to node in progress.


Copying files to node successful.
..................................................   79% Done.


Saving cluster inventory in progress.
..................................................   87% Done.


Saving cluster inventory successful.
The Cluster Node Addition of /u01/app/12.1.0/grid was
successful.
Please check '/tmp/silentInstall.log' for more details.


As a root user, execute the following script(s):
        1. /u01/app/12.1.0/grid/root.sh


Execute /u01/app/12.1.0/grid/root.sh on the following nodes:
[host05]


The scripts can be executed in parallel on all the nodes. If
there are any policy managed databases managed by cluster,
proceed with the addnode procedure without executing the root.sh
```

```
script. Ensure that root.sh script is executed after all the
policy managed databases managed by clusterware are extended to
the new nodes.
..........
Update Inventory in progress.
..................................................   100% Done.

Update Inventory successful.
Successfully Setup Software.
[grid@host01 addnode]$
```

5. Execute the `root.sh` script on `host05`.

```
[root@host01 ~]# ssh host05 /u01/app/12.1.0/grid/root.sh
Performing root user operation for Oracle 12c

The following environment variables are set as:
    ORACLE_OWNER= grid
    ORACLE_HOME=  /u01/app/12.1.0/grid
   Copying dbhome to /usr/local/bin ...
   Copying oraenv to /usr/local/bin ...
   Copying coraenv to /usr/local/bin ...


Creating /etc/oratab file...
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.
Relinking oracle with rac_on option
Using configuration parameter file:
/u01/app/12.1.0/grid/crs/install/crsconfig_params
2013/05/16 16:46:06 CLSRSC-363: User ignored prerequisites
during installation

OLR initialization - successful
2013/05/16 16:46:39 CLSRSC-330: Adding Clusterware entries to
file 'oracle-ohasd.conf'

CRS-4133: Oracle High Availability Services has been stopped.
CRS-4123: Oracle High Availability Services has been started.
CRS-4133: Oracle High Availability Services has been stopped.
CRS-4123: Oracle High Availability Services has been started.
```

```
CRS-2791: Starting shutdown of Oracle High Availability
Services-managed resources on 'host05'
CRS-2673: Attempting to stop 'ora.drivers.acfs' on 'host05'
CRS-2677: Stop of 'ora.drivers.acfs' on 'host05' succeeded
CRS-2793: Shutdown of Oracle High Availability Services-managed
resources on 'host05' has completed
CRS-4133: Oracle High Availability Services has been stopped.
CRS-4123: Starting Oracle High Availability Services-managed
resources
CRS-2672: Attempting to start 'ora.mdnsd' on 'host05'
CRS-2672: Attempting to start 'ora.evmd' on 'host05'
CRS-2676: Start of 'ora.mdnsd' on 'host05' succeeded
CRS-2676: Start of 'ora.evmd' on 'host05' succeeded
CRS-2672: Attempting to start 'ora.gpnpd' on 'host05'
CRS-2676: Start of 'ora.gpnpd' on 'host05' succeeded
CRS-2672: Attempting to start 'ora.gipcd' on 'host05'
CRS-2676: Start of 'ora.gipcd' on 'host05' succeeded
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'host05'
CRS-2676: Start of 'ora.cssdmonitor' on 'host05' succeeded
CRS-2672: Attempting to start 'ora.cssd' on 'host05'
CRS-2672: Attempting to start 'ora.diskmon' on 'host05'
CRS-2676: Start of 'ora.diskmon' on 'host05' succeeded
CRS-2789: Cannot stop resource 'ora.diskmon' as it is not
running on server 'host05'
CRS-2676: Start of 'ora.cssd' on 'host05' succeeded
CRS-2672: Attempting to start 'ora.cluster_interconnect.haip' on
'host05'
CRS-2672: Attempting to start 'ora.ctssd' on 'host05'
CRS-2676: Start of 'ora.cluster_interconnect.haip' on 'host05'
succeeded
CRS-2676: Start of 'ora.ctssd' on 'host05' succeeded
CRS-2672: Attempting to start 'ora.storage' on 'host05'
CRS-2676: Start of 'ora.storage' on 'host05' succeeded
CRS-2672: Attempting to start 'ora.crf' on 'host05'
CRS-2676: Start of 'ora.crf' on 'host05' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'host05'
CRS-2676: Start of 'ora.crsd' on 'host05' succeeded
CRS-6017: Processing resource auto-start for servers: host05
CRS-6016: Resource auto-start has completed for server host05
CRS-6024: Completed start of Oracle Cluster Ready Services-
managed resources
CRS-4123: Oracle High Availability Services has been started.
2013/05/16 16:50:35 CLSRSC-343: Successfully started Oracle
clusterware stack
```

```
Successfully accumulated necessary OCR keys.
Creating OCR keys for user 'root', privgrp 'root'..
Operation successful.
2013/05/16 16:50:41 CLSRSC-325: Configure Oracle Grid
Infrastructure for a Cluster ... succeeded

[root@host01 ~]#
```

6. Execute `crsctl stat res -t` and ensure the `ora.LISTENER_LEAF.lsnr` resource is located on `host05`. The resource status should be OFFLINE.

```
[root@host01 ~]# . oraenv
ORACLE_SID = [root] ? +ASM1
The Oracle base has been set to /u01/app/grid
[root@host01 ~]# crsctl stat res -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.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.LISTENER_LEAF.lsnr
               OFFLINE OFFLINE      host04                 STABLE
               OFFLINE OFFLINE      host05                 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       host03                STABLE
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       host01                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       host03                STABLE
ora.scan3.vip
      1        ONLINE  ONLINE       host01                STABLE
----------------------------------------------------------------
```

```
[root@host01 ~]#
```

7. At this point you have configured a Flex Cluster with Flex ASM using all available nodes. 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). Exit out of the grid terminal session and start s grid terminal session using the **`-X`** option. Start the ASM Configuration Assistant (`asmca`).

```
[grid@host01 addnode]$ exit (From the current grid terminal)

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

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

[grid@host01 ~]$ asmca
```

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

9. 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.

10. 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.

11. Click Exit to quit the ASM Configuration Assistant.

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

Practice 6-3: Installing RAC Database Software and Creating a RAC Database

Overview

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

1. Establish an `ssh` connection to `host01` using the `–X` option as the `oracle` user.

```
[vncuser@classsroom_pc ~]$ ssh -X oracle@host01
oracle@host01's password:
Last login: Tue Apr 30 17:07:09 2013 from 192.0.2.1
[oracle@host01 ~]$
```

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

```
[oracle@host01 ~]$ cd /stage/database
[oracle@host01 database]$ ./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 Configure Security Updates screen, make sure that The I wish to receive security updates via My Oracle Support check box is unselected 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, ensure that all the cluster nodes are selected and click SSH Connectivity. Note that Oracle recommends that you install the Oracle Database software on all the cluster nodes, even Leaf Nodes. This simplifies things if you ever want to convert a Leaf Node to a Hub Node and run database instances on it.

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. Execute the `orainstRoot.sh` script on `host05`, if prompted. 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`, `host03`, `host04` and `host05` 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
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
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 host04
/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.

[root@host01 ~]# ssh host05
/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.
[root@host01 ~]#
```

18. After you have executed the required configuration script on all of the 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 6-4: 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 database]$ 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 sure storage Type is ASM and make the following adjustments:

   - Database File Locations: +DATA
   - Fast Recovery Area: +FRA
   - Fast Recovery Area Size: 5400

   Space constraints in the laboratory environment require that you ignore the warning regarding the size of the Fast Recovery Area. Once your values match the values above, 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 12c Flex Cluster with Flex ASM and a RAC database.

Practices for Lesson 7: Traditional Clusterware Management

Chapter 7

Practices for Lesson 7: Overview

Practices Overview

In these practices, you will:

- Verify, stop, and start Oracle Clusterware.
- Add and remove Oracle Clusterware configuration files and back up the Oracle Cluster Registry and the Oracle Local Registry.
- Determine the location of the Oracle Local Registry (OLR) and perform backups of the OCR and OLR.
- Perform what-if command evaluation using the different commands for various different resources and examine the output formatting options available with what-if command evaluation.
- Use `oifcfg` to configure a second private interconnect.

Practice 7-1: Verifying, Starting, and Stopping Oracle Clusterware

Overview

In this practice, you check the status of Oracle Clusterware using both the operating system commands and the `crsctl` utility. You will also start and stop Oracle Clusterware.

1. Connect to the first node of your cluster as the `grid` user. You can use the `oraenv` script to define `ORACLE_SID`, `ORACLE_HOME`, `PATH`, `ORACLE_BASE`, and `LD_LIBRARY_PATH` for your environment.

```
[vncuser@classroom_pc ~] ssh grid@host01
Password: <oracle>

[grid@host01~]$ id
uid=54322(grid) gid=54321(oinstall)
groups=54321(oinstall),54327(asmdba),54328(asmoper),54329(asmadmin)
[grid@host01~] $ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base has been set to /u01/app/grid
```

2. Using operating system commands, verify that the Oracle Clusterware daemon processes are running on the current node. (Hint: Most of the Oracle Clusterware daemon processes have names that end with `d.bin`.)

```
[grid@host01~]$ pgrep -l d.bin
1743 ohasd.bin
1972 evmd.bin
1974 mdnsd.bin
1987 gpnpd.bin
2008 gipcd.bin
2204 ocssd.bin
2334 octssd.bin
2587 crsd.bin
2879 gnsd.bin
[grid@host01 ~]$
```

3. Using the `crsctl` utility, verify that Oracle Clusterware is running on the current node.

```
[grid@host01 ~]$ crsctl check crs
CRS-4638: Oracle High Availability Services is online
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
[grid@host01 ~]$
```

4. Verify the status of all cluster resources that are being managed by Oracle Clusterware for all nodes.

```
[grid@host01 ~]$ crsctl stat res -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.DATA.dg
               ONLINE  ONLINE       host01                   STABLE
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
ora.FRA.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.LISTENER_LEAF.lsnr
               OFFLINE OFFLINE      host04                   STABLE
               OFFLINE OFFLINE      host05                   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       host03                   STABLE
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       host01                   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       host03                   STABLE
ora.scan3.vip
      1        ONLINE  ONLINE       host01                   STABLE
--------------------------------------------------------------------
[grid@host01 ~]$
```

5. Attempt to stop Oracle Clusterware on the current node while logged in as the `grid` user. What happens and why?

```
[grid@host01 ~]$ crsctl stop crs
CRS-4563: Insufficient user privileges.
CRS-4000: Command Stop failed, or completed with errors.
[grid@host01 ~]$
```

6. Switch to the `root` account, set the environment with `oraenv` and stop Oracle Clusterware only on the current node.

```
[grid@host01 ~]$ su -
[root@host01 ~]# . oraenv
ORACLE_SID = [root] ? +ASM1
The Oracle base has been set to /u01/app/grid
[root@host01 ~]# crsctl stop crs
CRS-2791: Starting shutdown of Oracle High Availability Services-
managed resources on 'host01'
CRS-2673: Attempting to stop 'ora.crsd' on 'host01'
CRS-2790: Starting shutdown of Cluster Ready Services-managed
resources on 'host01'
CRS-2673: Attempting to stop 'ora.FRA.dg' on 'host01'
CRS-2673: Attempting to stop 'ora.DATA.dg' on 'host01'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on 'host01'
CRS-2673: Attempting to stop 'ora.LISTENER_SCAN3.lsnr' on 'host01'
CRS-2673: Attempting to stop 'ora.cvu' on 'host01'
CRS-2673: Attempting to stop 'ora.oc4j' on 'host01'
CRS-2673: Attempting to stop 'ora.proxy_advm' on 'host01'
CRS-2673: Attempting to stop 'ora.gns' on 'host01'
CRS-2677: Stop of 'ora.LISTENER.lsnr' on 'host01' succeeded
CRS-2677: Stop of 'ora.LISTENER_SCAN3.lsnr' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.scan3.vip' on 'host01'
CRS-2673: Attempting to stop 'ora.host01.vip' on 'host01'
CRS-2677: Stop of 'ora.FRA.dg' on 'host01' succeeded
CRS-2677: Stop of 'ora.scan3.vip' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.scan3.vip' on 'host02'
CRS-2677: Stop of 'ora.DATA.dg' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host01'
CRS-2677: Stop of 'ora.host01.vip' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.host01.vip' on 'host03'
CRS-2676: Start of 'ora.scan3.vip' on 'host02' succeeded
CRS-2676: Start of 'ora.host01.vip' on 'host03' succeeded
CRS-2677: Stop of 'ora.asm' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.ASMNET1LSNR_ASM.lsnr' on 'host01'
CRS-2677: Stop of 'ora.cvu' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.cvu' on 'host03'
CRS-2672: Attempting to start 'ora.LISTENER_SCAN3.lsnr' on 'host02'
CRS-2676: Start of 'ora.cvu' on 'host03' succeeded
CRS-2677: Stop of 'ora.gns' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.gns.vip' on 'host01'
CRS-2677: Stop of 'ora.gns.vip' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.gns.vip' on 'host02'
```

```
CRS-2676: Start of 'ora.gns.vip' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.gns' on 'host02'
CRS-2677: Stop of 'ora.ASMNET1LSNR_ASM.lsnr' on 'host01' succeeded
CRS-2676: Start of 'ora.LISTENER_SCAN3.lsnr' on 'host02' succeeded
CRS-2677: Stop of 'ora.proxy_advm' on 'host01' succeeded
CRS-2677: Stop of 'ora.oc4j' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.oc4j' on 'host02'
CRS-2676: Start of 'ora.gns' on 'host02' succeeded
CRS-2676: Start of 'ora.oc4j' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.ons' on 'host01'
CRS-2677: Stop of 'ora.ons' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.net1.network' on 'host01'
CRS-2677: Stop of 'ora.net1.network' on 'host01' succeeded
CRS-2792: Shutdown of Cluster Ready Services-managed resources on
'host01' has completed
CRS-2677: Stop of 'ora.crsd' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.crf' on 'host01'
CRS-2673: Attempting to stop 'ora.ctssd' on 'host01'
CRS-2673: Attempting to stop 'ora.evmd' on 'host01'
CRS-2673: Attempting to stop 'ora.storage' on 'host01'
CRS-2673: Attempting to stop 'ora.gpnpd' on 'host01'
CRS-2673: Attempting to stop 'ora.mdnsd' on 'host01'
CRS-2673: Attempting to stop 'ora.drivers.acfs' on 'host01'
CRS-2677: Stop of 'ora.storage' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host01'
CRS-2677: Stop of 'ora.drivers.acfs' on 'host01' succeeded
CRS-2677: Stop of 'ora.crf' on 'host01' succeeded
CRS-2677: Stop of 'ora.gpnpd' on 'host01' succeeded
CRS-2677: Stop of 'ora.mdnsd' on 'host01' succeeded
CRS-2677: Stop of 'ora.evmd' on 'host01' succeeded
CRS-2677: Stop of 'ora.ctssd' on 'host01' succeeded
CRS-2677: Stop of 'ora.asm' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.cluster_interconnect.haip' on
'host01'
CRS-2677: Stop of 'ora.cluster_interconnect.haip' on 'host01'
succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host01'
CRS-2677: Stop of 'ora.cssd' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.gipcd' on 'host01'
CRS-2677: Stop of 'ora.gipcd' on 'host01' succeeded
CRS-2793: Shutdown of Oracle High Availability Services-managed
resources on 'host01' has completed
CRS-4133: Oracle High Availability Services has been stopped.
```

```
[root@host01 ~]#
```

7. Attempt to check the status of Oracle Clusterware now that it has been successfully stopped.

```
[root@host01 ~]# crsctl check cluster
CRS-4639: Could not contact Oracle High Availability Services
CRS-4000: Command Check failed, or completed with errors.

[root@host01 ~]#
```

8. Connect to the second node of your cluster and verify that Oracle Clusterware is still running on that node. Set your environment for the second node by using the `oraenv` utility.

```
[root@host01 ~]# ssh host02
Last login: Thu May 16 16:00:25 2013 from host01.example.com
[root@host02 ~]# . oraenv
ORACLE_SID = [root] ? +ASM2
The Oracle base has been set to /u01/app/grid
[root@host02 ~]# crsctl check crs
CRS-4638: Oracle High Availability Services is online
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
[root@host02 ~]#
```

9. Verify that all cluster resources are running on the second and third nodes, stopped on the first node, and that the VIP resources from the first node have migrated or failed over to the other two nodes. When you are finished, exit from `host02`.

```
[root@host02 ~]# crsctl stat res -t
--------------------------------------------------------------------------------
Name           Target  State        Server                   State details
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.ASMNET1LSNR_ASM.lsnr
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
ora.DATA.dg
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
ora.FRA.dg
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
ora.LISTENER.lsnr
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
ora.LISTENER_LEAF.lsnr
               OFFLINE OFFLINE      host04                   STABLE
               OFFLINE OFFLINE      host05                   STABLE
ora.net1.network
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
ora.ons
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
ora.proxy_advm
               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       host03                   STABLE
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       host02                   STABLE
ora.asm
      1        ONLINE  OFFLINE                               STABLE
      2        ONLINE  ONLINE       host02                   STABLE
      3        ONLINE  ONLINE       host03                   STABLE
ora.cvu
      1        ONLINE  ONLINE       host03                   STABLE
ora.gns
      1        ONLINE  ONLINE       host02                   STABLE
ora.gns.vip
      1        ONLINE  ONLINE       host02                   STABLE
ora.host01.vip
      1        ONLINE  INTERMEDIATE host03       FAILED OVER,STABLE
ora.host02.vip
      1        ONLINE  ONLINE       host02                   STABLE
ora.host03.vip
      1        ONLINE  ONLINE       host03                   STABLE
ora.oc4j
```

```
      1        ONLINE  ONLINE       host02                   STABLE
ora.scan1.vip
      1        ONLINE  ONLINE       host02                   STABLE
ora.scan2.vip
      1        ONLINE  ONLINE       host03                   STABLE
ora.scan3.vip
      1        ONLINE  ONLINE       host02                   STABLE
-------------------------------------------------------------------
[root@host02 ~]# exit
logout
Connection to host02 closed.
[root@host01 ~]#
```

10. Restart Oracle Clusterware on `host01` as the `root` user. Exit from the `root` account back to `grid` and verify the results of the `crsctl start crs` command you just executed.

```
[root@host01 ~]# crsctl start crs
CRS-4123: Oracle High Availability Services has been started.

[root@host01 ~]# exit
logout
Connection to host01 closed.

[grid@host01 ~]$ crsctl stat res -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.DATA.dg
               ONLINE  ONLINE       host01                   STABLE
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
ora.FRA.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.LISTENER_LEAF.lsnr
               OFFLINE OFFLINE      host04                   STABLE
               OFFLINE OFFLINE      host05                   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       host01                   STABLE
ora.LISTENER_SCAN2.lsnr
      1        ONLINE  ONLINE       host03                   STABLE
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       host02                   STABLE
ora.asm
      1        ONLINE  ONLINE       host01                   STABLE
      2        ONLINE  ONLINE       host02                   STABLE
      3        ONLINE  ONLINE       host03                   STABLE
ora.cvu
      1        ONLINE  ONLINE       host03                   STABLE
ora.gns
      1        ONLINE  ONLINE       host02                   STABLE
ora.gns.vip
      1        ONLINE  ONLINE       host02                   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       host02                   STABLE
ora.scan1.vip
      1        ONLINE  ONLINE       host01                   STABLE
ora.scan2.vip
      1        ONLINE  ONLINE       host03                   STABLE
ora.scan3.vip
      1        ONLINE  ONLINE       host02                   STABLE
--------------------------------------------------------------------
[grid@host01 ~]$
```

Practice 7-2: Adding and Removing Oracle Clusterware Configuration Files

Overview

In this practice, you determine the current location of your voting disks and Oracle Cluster Registry (OCR) files. You will then add another OCR location and remove it.

1. As the `grid` user, use the `crsctl` utility to determine the location of the voting disks that are currently used by your Oracle Clusterware installation.

```
[grid@host01 ~]$ crsctl query css votedisk
##  STATE    File Universal Id                File Name Disk group
--  -----    -----------------                --------- ---------
 1. ONLINE   7b50c15f90dd4f8bbf81d40d84f81f33 (/dev/asmdisk1p1)
[DATA]
 2. ONLINE   d084edbe14444ffdbf362ff8ec50c511 (/dev/asmdisk1p10)
[DATA]
 3. ONLINE   3c7041d9c84e4ffdbfc550f315b61226 (/dev/asmdisk1p11)
[DATA]
Located 3 voting disk(s).
[grid@host01 ~]$
```

2. Use the `ocrcheck` utility to determine the location of the Oracle Clusterware Registry (OCR) files.

```
[grid@host01 ~]$ ocrcheck -config
Oracle Cluster Registry configuration is :
         Device/File Name         :      +DATA
[grid@host01 ~]$
```

3. Verify that the `FRA` ASM disk group is currently online for all nodes using the `crsctl` utility.

```
[grid@host01 ~]$ crsctl stat res ora.FRA.dg -t
----------------------------------------------------------------
Name           Target  State        Server                State
details
----------------------------------------------------------------
Local Resources
---------------------------------------------------------------
ora.FRA.dg
               ONLINE  ONLINE       host01                STABLE
               ONLINE  ONLINE       host02                STABLE
               ONLINE  ONLINE       host03                STABLE

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

4. Switch to the `root` account and add a second OCR location that is to be stored in the `FRA` ASM disk group. Set the environment with `oraenv` and use the `ocrcheck` command to verify the results.

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

[root@host01 ~]# ocrconfig -add +FRA

[[root@host01 ~]# ocrcheck -config
Oracle Cluster Registry configuration is :
         Device/File Name         :      +DATA
         Device/File Name         :       +FRA
[root@host01 ~]#
```

5. Examine the contents of the `ocr.loc` configuration file to see the changes made to the file referencing the new OCR location.

```
[root@host01 ~]# cat /etc/oracle/ocr.loc
#Device/file  getting replaced by device +FRA
ocrconfig_loc=+DATA
ocrmirrorconfig_loc=+FRA
local_only=false[root@host01 ~]#
```

6. Open a connection to your second node as the `root` user, set your environment and remove the second OCR file that was added from the first node. Exit the remote connection and verify the results when completed.

```
[root@host01 ~]# ssh host02
root@host02's password:
[root@host02 ~]# . oraenv
ORACLE_SID = [root] ? +ASM2
The Oracle base has been set to /u01/app/grid
[root@host02 ~]# ocrconfig -delete +FRA
[root@host02 ~]# exit
logout
Connection to host02 closed.

[root@host01 ~]# ocrcheck -config
Oracle Cluster Registry configuration is :
         Device/File Name         :      +DATA
```

```
[root@host01 ~]#
```

Practice 7-3: Performing a Backup of the OCR and OLR

Overview

In this practice, you determine the location of the Oracle Local Registry (OLR) and perform backups of the OCR and OLR files.

1. As the `root` user, use the `ocrconfig` utility to list the automatic backups of the Oracle Cluster Registry (OCR) and the node or nodes on which they have been performed.

   Note: You will see backups listed only if it has been more than four hours since Grid Infrastructure was installed. Your list will most like look slightly different from the example below.

```
[root@host01 ~]# ocrconfig -showbackup

host01     2013/09/19 08:07:28
/u01/app/12.1.0/grid/cdata/cluster01/backup00.ocr

host01     2013/09/19 04:07:26
/u01/app/12.1.0/grid/cdata/cluster01/backup01.ocr

host01     2013/09/19 00:07:25
/u01/app/12.1.0/grid/cdata/cluster01/backup02.ocr

host01     2013/09/19 00:07:25
/u01/app/12.1.0/grid/cdata/cluster01/day.ocr

host01     2013/09/19 00:07:25
/u01/app/12.1.0/grid/cdata/cluster01/week.ocr

host01     2013/09/19 05:50:54
/u01/app/12.1.0/grid/cdata/cluster01/backup_20130919_055054.ocr

host01     2013/09/19 04:47:54
/u01/app/12.1.0/grid/cdata/cluster01/backup_20130919_044754.ocr

host01     2013/09/18 20:08:18
/u01/app/12.1.0/grid/cdata/cluster01/backup_20130918_200818.ocr

host01     2013/09/18 20:01:51
/u01/app/12.1.0/grid/cdata/cluster01/backup_20130918_200151.ocr

host01     2013/09/18 19:54:43
/u01/app/12.1.0/grid/cdata/cluster01/backup_20130918_195443.ocr
[root@host01 ~]#
```

2. Perform a manual backup of the OCR.

```
[root@host01 ~]# ocrconfig -manualbackup

host02     2013/05/20 11:16:09
/u01/app/12.1.0/grid/cdata/cluster01/backup_20130520_111609.ocr

host01     2013/05/16 11:20:14
/u01/app/12.1.0/grid/cdata/cluster01/backup_20130516_112014.ocr

host01     2013/05/07 20:38:04
/u01/app/12.1.0/grid/cdata/cluster01/backup_20130507_203804.ocr

[root@host01 ~]#
```

Note: Your output may vary slightly.

3. Use the `crsctl` utility to determine the OCR locations that are currently used by your Oracle Clusterware installation.

```
[root@host01 ~]# ocrconfig -showbackup manual

host02     2013/09/19 08:49:32
/u01/app/12.1.0/grid/cdata/cluster01/backup_20130919_084932.ocr

host01     2013/09/19 05:50:54
/u01/app/12.1.0/grid/cdata/cluster01/backup_20130919_055054.ocr

host01     2013/09/19 04:47:54
/u01/app/12.1.0/grid/cdata/cluster01/backup_20130919_044754.ocr

host01     2013/09/18 20:08:18
/u01/app/12.1.0/grid/cdata/cluster01/backup_20130918_200818.ocr

host01     2013/09/18 20:01:51
/u01/app/12.1.0/grid/cdata/cluster01/backup_20130918_200151.ocr
[root@host01 ~]#
```

4. Determine the location of the Oracle Local Registry (OLR) using the `ocrcheck` utility. When finished, log out as the `root` user.

```
[root@host01 ~]# ocrcheck -local
Status of Oracle Local Registry is as follows :
   Version                  :          4
   Total space (kbytes)     :     409568
   Used space (kbytes)      :       1056
   Available space (kbytes) :     408512
   ID                       : 1935905403
   Device/File Name         : /u01/app/12.1.0/grid/cdata/host01.olr
                              Device/File integrity check succeeded

   Local registry integrity check succeeded

   Logical corruption check succeeded

[root@host01 ~]# exit
logout

[grid@host01 ~]$
```

Practice 7-4: Using What-If Command Evaluation

Overview

In this practice you perform what-if command evaluation using the different commands for various different resources. You will also examine the different output formatting options available with what-if command evaluation.

1. Check the status of your cluster resources using the `crsctl` command.

```
[grid@host01 ~]$ crsctl stat res -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.DATA.dg
               ONLINE  ONLINE       host01                   STABLE
               ONLINE  ONLINE       host02                   STABLE
               ONLINE  ONLINE       host03                   STABLE
ora.FRA.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.LISTENER_LEAF.lsnr
               OFFLINE OFFLINE      host04                   STABLE
               OFFLINE OFFLINE      host05                   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       host01                   STABLE
ora.LISTENER_SCAN2.lsnr
      1        ONLINE  ONLINE       host03                   STABLE
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       host02                   STABLE
ora.asm
      1        ONLINE  ONLINE       host01                   STABLE
      2        ONLINE  ONLINE       host02                   STABLE
      3        ONLINE  ONLINE       host03                   STABLE
ora.cvu
      1        ONLINE  ONLINE       host03                   STABLE
ora.gns
      1        ONLINE  ONLINE       host02                   STABLE
ora.gns.vip
      1        ONLINE  ONLINE       host02                   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       host02                   STABLE
ora.scan1.vip
      1        ONLINE  ONLINE       host01                   STABLE
ora.scan2.vip
      1        ONLINE  ONLINE       host03                   STABLE
ora.scan3.vip
      1        ONLINE  ONLINE       host02                   STABLE
--------------------------------------------------------------------
[grid@host01 ~]$
```

2. The `ora.cvu` cluster resource is placed on a single node in the cluster. In the example above, it is located on `host03`. Use the `crsctl eval` command to evaluate the impact of stopping an innocuous resource like `ora.cvu` .

```
[grid@host01 ~]$ crsctl eval stop resource ora.cvu -f

Stage Group 1:
--------------------------------------------------------------------
Stage Number  Required   Action
--------------------------------------------------------------------

    1             Y      Resource 'ora.cvu' (1/1) will be in state
                         [OFFLINE]

--------------------------------------------------------------------

[grid@host01 ~]$
```

3. Use the `crsctl eval start resource` command to evaluate the impact of starting the `ora.cvu` resource on a different host, in this example it is `host01`. As the `ora.cvu` is a singleton resource and already running on another host, the `crsctl eval` command reports this

```
[grid@host01 ~]$ crsctl eval start resource ora.cvu -n host01 -f

Stage Group 1:
--------------------------------------------------------------------
Stage Number   Required   Action
--------------------------------------------------------------------

   1            N         Error code [223] for entity [ora.cvu].
                          Message is [CRS-2552: There are no
                          available instances of resource 'ora.cvu'
                          to start.].

--------------------------------------------------------------------

[grid@host01 ~]$
```

4. Stop the ora.cvu resource using the `-f` option and rerun the `crsctl eval` command used in the previous step.

```
[grid@host01 ~]$ crsctl stop resource ora.cvu -f
CRS-2673: Attempting to stop 'ora.cvu' on 'host03'
CRS-2677: Stop of 'ora.cvu' on 'host03' succeeded

[grid@host01 ~]$ crsctl eval start resource ora.cvu -n host01 -f

Stage Group 1:
--------------------------------------------------------------------
Stage Number   Required    Action
--------------------------------------------------------------------


     1              Y     Resource 'ora.cvu' (1/1) will be in state
                          [ONLINE|INTERMEDIATE] on server [host01]


--------------------------------------------------------------------


[grid@host01 ~]$
```

5. Start the `ora.cvu` resource with the `-f` option. Check the result with `crsctl stat res` command and check the results with the `crsctl eval` result from the previous step.

```
[grid@host01 ~]$ crsctl start resource ora.cvu -n host01 -f
CRS-2672: Attempting to start 'ora.cvu' on 'host01'
CRS-2676: Start of 'ora.cvu' on 'host01' succeeded
[grid@host01 ~]$
[grid@host01 ~]$ crsctl stat res -t -w "NAME co ora.cvu"
--------------------------------------------------------------------
Name           Target  State        Server                    State
details
--------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------
ora.cvu
      1        ONLINE  ONLINE       host01                    STABLE
--------------------------------------------------------------------
[grid@host01 ~]$
```

6. Next, use what-if command evaluation to view the impact of stopping a critical, local resource like `ora.asm`. Do ***not*** stop the `ora.asm` resource.

```
[grid@host01 ~]$ crsctl eval stop resource ora.asm -f

Stage Group 1:
--------------------------------------------------------------------
Stage Number Required    Action
--------------------------------------------------------------------


     1          Y        Resource 'ora.DATA.dg' (host01) will be in
                         state [OFFLINE]
                Y        Resource 'ora.DATA.dg' (host02) will be in
                         state [OFFLINE]
                Y        Resource 'ora.DATA.dg' (host03) will be in
                         state [OFFLINE]
                Y        Resource 'ora.FRA.dg' (host01) will be in
                         state [OFFLINE]
                Y        Resource 'ora.FRA.dg' (host02) will be in
                         state [OFFLINE]
                Y        Resource 'ora.FRA.dg' (host03) will be in
                         state [OFFLINE]
                Y        Resource 'ora.proxy_advm' (host01) will be
                         in state [OFFLINE]
                Y        Resource 'ora.proxy_advm' (host02) will be
                         in state [OFFLINE]
                Y        Resource 'ora.proxy_advm' (host03) will be
                         In state [OFFLINE]

     2          Y        Resource 'ora.asm' (1/1) will be in state
                         [OFFLINE]
                Y        Resource 'ora.asm' (2/1) will be in state
                         [OFFLINE]
                Y        Resource 'ora.asm' (3/1) will be in state
                         [OFFLINE]


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

7. Now use what-if command evaluation to analyze the effect of removing the server `host02` from the cluster. Notice how the output is divided into stages. In this example, the actions in stage one all relate to the immediate effects of removing the server. This includes removing `host02` from its current server pool and various resources stopping because the server is no longer available.

   In stage two you can see how the VIP resources associated with `host02` are migrated to another server. Finally, in stage three, a SCAN listener resource is also migrated. The migration of the SCAN listener resource occurs last and appears in a separate stage because that resource depends on one of the migrated VIP resources.

```
[grid@host01 ~]$ crsctl eval delete server host02 -f

Stage Group 1:
--------------------------------------------------------------------
Stage Number   Required    Action
--------------------------------------------------------------------


    1            Y         Server 'host02' will be removed from
                           Pools [Free]
                 Y         Resource 'ora.ASMNET1LSNR_ASM.lsnr'
                           (host02) will be in state [OFFLINE]
                 Y         Resource 'ora.DATA.dg' (host02) will be
                           in state [OFFLINE]
                 Y         Resource 'ora.FRA.dg' (host02) will be
                           in state [OFFLINE]
                 Y         Resource 'ora.LISTENER.lsnr' (host02)
                           will be in state [OFFLINE]
                 Y         Resource 'ora.LISTENER_SCAN3.lsnr' (1/1)
                           will be in state [OFFLINE]
                 Y         Resource 'ora.asm' (2/1) will be in
                           state [OFFLINE]
                 Y         Resource 'ora.gns' (1/1) will be in
                           state [OFFLINE]
                 Y         Resource 'ora.gns.vip' (1/1) will be in
                           state [OFFLINE]
                 Y         Resource 'ora.host02.vip' (1/1) will be
                           in state [OFFLINE]
                 Y         Resource 'ora.net1.network' (host02)
                           will be in state [OFFLINE]
                 Y         Resource 'ora.oc4j' (1/1) will be in
                           state [OFFLINE]
                 Y         Resource 'ora.ons' (host02) will be in
                           state [OFFLINE]
                 Y         Resource 'ora.proxy_advm' (host02) will
```

```
                           be in state [OFFLINE]
                 Y         Resource 'ora.scan3.vip' (1/1) will be
                           in state [OFFLINE]

     2           Y         Resource 'ora.gns.vip' (1/1) will be in
                           state [ONLINE] on server [host03]
                 Y         Resource 'ora.oc4j' (1/1) will be in
                           state [ONLINE|INTERMEDIATE] on server
                           [host03]

     3           N         Resource 'ora.gns' (1/1) will be in
                           state [ONLINE|INTERMEDIATE] on server
                           [host01]

     4           Y         Resource 'ora.host02.vip' (1/1) will be
                           in state [ONLINE|INTERMEDIATE] on server
                           [host03]
                 Y         Resource 'ora.scan3.vip' (1/1) will be
                           in state [ONLINE] on server [host01]

     5           Y         Resource 'ora.LISTENER_SCAN3.lsnr' (1/1)
                           will be in state [ONLINE|INTERMEDIATE]
                           on server [host01]

-------------------------------------------------------------------

[grid@host01 ~]$
```

Practice 7-5: Configuring Network Interfaces Using `oifcfg`

Overview

In this practice, you add a new interface to create an HAIP configuration for the cluster.

1. As the `grid` user, ensure that Oracle Clusterware is running on all of the cluster nodes.

```
[grid@host01 ~]$ olsnodes -s
host01  Active
host02  Active
host03  Active
host04  Active
host05  Active
[grid@host01 ~]$
```

2. Check the list of available network interfaces and ascertain what they are being used for in the current cluster configuration

```
[grid@host01 ~]$ oifcfg iflist
eth0  192.0.2.0
eth1  192.168.1.0
eth1  169.254.0.0
eth2  192.168.2.0

[grid@host01 ~]$ oifcfg getif
eth0  192.0.2.0  global  public
eth1  192.168.1.0  global  cluster_interconnect,asm
[grid@host01 ~]$
```

3. The `eth0` interface is currently used for the public interface and `eth1` is used for the private interconnect/ASM. We'll add `eth2` to the interconnect configuration. Ensure that the replacement interface is configured and operational in the operating system on all of the nodes.

```
[grid@host01 ~]$ /sbin/ifconfig eth2
eth2 Link encap:Ethernet  HWaddr 00:16:3E:00:01:21
     inet addr: 192.168.2.101 Bcast:192.168.2.255 Mask:255.255.255.0
     inet6 addr: fe80::216:3eff:fe00:121/64 Scope:Link
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:240917 errors:0 dropped:0 overruns:0 frame:0
     TX packets:30185 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:1000
     RX bytes:37543266 (35.8 MiB)  TX bytes:3499615 (3.3 MiB)
     Interrupt:35

[grid@host01 ~]$ ssh host02 /sbin/ifconfig eth2
eth2 Link encap:Ethernet  HWaddr 00:16:3E:00:01:22
     inet addr:192.168.2.102 Bcast:192.168.2.255  Mask:255.255.255.0
     inet6 addr: fe80::216:3eff:fe00:122/64 Scope:Link
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:232912 errors:0 dropped:0 overruns:0 frame:0
     TX packets:30491 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:1000
     RX bytes:36034120 (34.3 MiB)  TX bytes:3454982 (3.2 MiB)
     Interrupt:35

[grid@host01 ~]$ ssh host03 /sbin/ifconfig eth2
eth2 Link encap:Ethernet  HWaddr 00:16:3E:00:01:23
     inet addr:192.168.2.103 Bcast:192.168.2.255  Mask:255.255.255.0
     inet6 addr: fe80::216:3eff:fe00:123/64 Scope:Link
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:351430 errors:0 dropped:0 overruns:0 frame:0
     TX packets:29898 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:1000
     RX bytes:54946101 (52.4 MiB)  TX bytes:3424947 (3.2 MiB)
     Interrupt:35

[grid@host01 ~]$ ssh host04 /sbin/ifconfig eth2
eth2 Link encap:Ethernet  HWaddr 00:16:3E:00:01:24
     inet addr:192.168.2.104 Bcast:192.168.2.255  Mask:255.255.255.0
     inet6 addr: fe80::216:3eff:fe00:124/64 Scope:Link
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:49993 errors:0 dropped:0 overruns:0 frame:0
     TX packets:7194 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:1000
     RX bytes:7597687 (7.2 MiB)  TX bytes:821526 (802.2 KiB)
          Interrupt:33

[grid@host01 ~]$ ssh host05 /sbin/ifconfig eth2
eth2 Link encap:Ethernet  HWaddr 00:16:3E:00:01:25
     inet addr:192.168.2.105 Bcast:192.168.2.255  Mask:255.255.255.0
     inet6 addr: fe80::216:3eff:fe00:125/64 Scope:Link
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:323700 errors:0 dropped:0 overruns:0 frame:0
     TX packets:35817 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:1000
     RX bytes:50238663 (47.9 MiB)  TX bytes:3758667 (3.5 MiB)
     Interrupt:33

[grid@host01 ~]$
```

4. Switch user to `root`, set the Grid environment, and add `eth2` to the cluster configuration as a global private interconnect and ASM.

```
[grid@host01 ~]$ su -
Password:

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

[root@host01 ~]# oifcfg setif -global
eth2/192.168.2.0:cluster_interconnect,asm
```

```

[root@host01 ~]#
```

5. Verify that the new interface has been added to the cluster configuration.

```
[root@host01 ~]$ oifcfg getif
eth0  192.0.2.0  global  public
eth1  192.168.1.0  global  cluster_interconnect,asm
eth2  192.168.2.0  global  cluster_interconnect,asm
[root@host01 ~]$
```

6. Stop Clusterware on all nodes of the cluster.

```
[root@host01 ~]# crsctl stop cluster -all
CRS-2673: Attempting to stop 'ora.crsd' on 'host04'
CRS-2673: Attempting to stop 'ora.crsd' on 'host05'
CRS-2677: Stop of 'ora.crsd' on 'host04' succeeded
CRS-2673: Attempting to stop 'ora.cluster_interconnect.haip' on
'host04'
CRS-2673: Attempting to stop 'ora.ctssd' on 'host04'
CRS-2673: Attempting to stop 'ora.evmd' on 'host04'
CRS-2673: Attempting to stop 'ora.storage' on 'host04'
CRS-2677: Stop of 'ora.storage' on 'host04' succeeded
CRS-2677: Stop of 'ora.crsd' on 'host05' succeeded
CRS-2673: Attempting to stop 'ora.cluster_interconnect.haip' on
'host05'
CRS-2673: Attempting to stop 'ora.ctssd' on 'host05'
CRS-2673: Attempting to stop 'ora.evmd' on 'host05'
CRS-2673: Attempting to stop 'ora.storage' on 'host05'
CRS-2677: Stop of 'ora.storage' on 'host05' succeeded
CRS-2677: Stop of 'ora.cluster_interconnect.haip' on 'host04'
succeeded
CRS-2677: Stop of 'ora.cluster_interconnect.haip' on 'host05'
succeeded
CRS-2677: Stop of 'ora.evmd' on 'host04' succeeded
CRS-2677: Stop of 'ora.ctssd' on 'host04' succeeded
CRS-2677: Stop of 'ora.ctssd' on 'host05' succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host04'
CRS-2677: Stop of 'ora.cssd' on 'host04' succeeded
CRS-2677: Stop of 'ora.evmd' on 'host05' succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host05'
CRS-2677: Stop of 'ora.cssd' on 'host05' succeeded
CRS-2673: Attempting to stop 'ora.crsd' on 'host01'
CRS-2673: Attempting to stop 'ora.crsd' on 'host03'
CRS-2790: Starting shutdown of Cluster Ready Services-managed
resources on 'host01'
CRS-2673: Attempting to stop 'ora.cvu' on 'host01'
CRS-2673: Attempting to stop 'ora.FRA.dg' on 'host01'
CRS-2673: Attempting to stop 'ora.DATA.dg' on 'host01'
CRS-2673: Attempting to stop 'ora.orcl.db' on 'host01'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on 'host01'
CRS-2673: Attempting to stop 'ora.LISTENER_SCAN1.lsnr' on 'host01'
```

```
CRS-2673: Attempting to stop 'ora.proxy_advm' on 'host01'
CRS-2673: Attempting to stop 'ora.crsd' on 'host02'
CRS-2790: Starting shutdown of Cluster Ready Services-managed
resources on 'host02'
CRS-2790: Starting shutdown of Cluster Ready Services-managed
resources on 'host03'
CRS-2673: Attempting to stop 'ora.LISTENER_SCAN2.lsnr' on 'host03'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on 'host03'
CRS-2673: Attempting to stop 'ora.orcl.db' on 'host03'
CRS-2673: Attempting to stop 'ora.proxy_advm' on 'host03'
CRS-2673: Attempting to stop 'ora.orcl.db' on 'host02'
CRS-2673: Attempting to stop 'ora.LISTENER_SCAN3.lsnr' on 'host02'
CRS-2673: Attempting to stop 'ora.oc4j' on 'host02'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on 'host02'
CRS-2673: Attempting to stop 'ora.gns' on 'host02'
CRS-2673: Attempting to stop 'ora.proxy_advm' on 'host02'
CRS-2677: Stop of 'ora.LISTENER_SCAN1.lsnr' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.scan1.vip' on 'host01'
CRS-2677: Stop of 'ora.LISTENER.lsnr' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.host01.vip' on 'host01'
CRS-2677: Stop of 'ora.scan1.vip' on 'host01' succeeded
CRS-2677: Stop of 'ora.cvu' on 'host01' succeeded
CRS-2677: Stop of 'ora.host01.vip' on 'host01' succeeded
CRS-2677: Stop of 'ora.gns' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.gns.vip' on 'host02'
CRS-2677: Stop of 'ora.FRA.dg' on 'host01' succeeded
CRS-2677: Stop of 'ora.LISTENER.lsnr' on 'host03' succeeded
CRS-2673: Attempting to stop 'ora.host03.vip' on 'host03'
CRS-2677: Stop of 'ora.LISTENER_SCAN2.lsnr' on 'host03' succeeded
CRS-2673: Attempting to stop 'ora.scan2.vip' on 'host03'
CRS-2677: Stop of 'ora.gns.vip' on 'host02' succeeded
CRS-2677: Stop of 'ora.DATA.dg' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host01'
CRS-2677: Stop of 'ora.LISTENER_SCAN3.lsnr' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.scan3.vip' on 'host02'
CRS-2677: Stop of 'ora.scan2.vip' on 'host03' succeeded
CRS-2677: Stop of 'ora.LISTENER.lsnr' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.host02.vip' on 'host02'
CRS-2677: Stop of 'ora.host03.vip' on 'host03' succeeded
CRS-2677: Stop of 'ora.scan3.vip' on 'host02' succeeded
CRS-2677: Stop of 'ora.host02.vip' on 'host02' succeeded
CRS-2677: Stop of 'ora.orcl.db' on 'host01' succeeded
CRS-2677: Stop of 'ora.asm' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.ASMNET1LSNR_ASM.lsnr' on 'host01'
CRS-2677: Stop of 'ora.ASMNET1LSNR_ASM.lsnr' on 'host01' succeeded
CRS-2677: Stop of 'ora.proxy_advm' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.DATA.dg' on 'host03'
CRS-2673: Attempting to stop 'ora.ons' on 'host01'
CRS-2677: Stop of 'ora.ons' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.net1.network' on 'host01'
CRS-2677: Stop of 'ora.net1.network' on 'host01' succeeded
```

```
CRS-2792: Shutdown of Cluster Ready Services-managed resources on
'host01' has completed
CRS-2675: Stop of 'ora.oc4j' on 'host02' failed
CRS-2679: Attempting to clean 'ora.oc4j' on 'host02'
CRS-2681: Clean of 'ora.oc4j' on 'host02' succeeded
CRS-2677: Stop of 'ora.crsd' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.ctssd' on 'host01'
CRS-2673: Attempting to stop 'ora.evmd' on 'host01'
CRS-2673: Attempting to stop 'ora.storage' on 'host01'
CRS-2677: Stop of 'ora.storage' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host01'
CRS-2677: Stop of 'ora.ctssd' on 'host01' succeeded
CRS-2677: Stop of 'ora.evmd' on 'host01' succeeded
CRS-2677: Stop of 'ora.proxy_advm' on 'host02' succeeded
CRS-2677: Stop of 'ora.proxy_advm' on 'host03' succeeded
CRS-2677: Stop of 'ora.orcl.db' on 'host03' succeeded
CRS-2677: Stop of 'ora.DATA.dg' on 'host03' succeeded
CRS-2673: Attempting to stop 'ora.FRA.dg' on 'host03'
CRS-2677: Stop of 'ora.FRA.dg' on 'host03' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host03'
CRS-2677: Stop of 'ora.asm' on 'host03' succeeded
CRS-2673: Attempting to stop 'ora.ASMNET1LSNR_ASM.lsnr' on 'host03'
CRS-2677: Stop of 'ora.ASMNET1LSNR_ASM.lsnr' on 'host03' succeeded
CRS-2673: Attempting to stop 'ora.ons' on 'host03'
CRS-2677: Stop of 'ora.ons' on 'host03' succeeded
CRS-2673: Attempting to stop 'ora.net1.network' on 'host03'
CRS-2677: Stop of 'ora.net1.network' on 'host03' succeeded
CRS-2792: Shutdown of Cluster Ready Services-managed resources on
'host03' has completed
CRS-2677: Stop of 'ora.asm' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.cluster_interconnect.haip' on
'host01'
CRS-2677: Stop of 'ora.cluster_interconnect.haip' on 'host01'
succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host01'
CRS-2677: Stop of 'ora.cssd' on 'host01' succeeded
CRS-2677: Stop of 'ora.orcl.db' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.FRA.dg' on 'host02'
CRS-2673: Attempting to stop 'ora.DATA.dg' on 'host02'
CRS-2677: Stop of 'ora.crsd' on 'host03' succeeded
CRS-2673: Attempting to stop 'ora.ctssd' on 'host03'
CRS-2673: Attempting to stop 'ora.evmd' on 'host03'
CRS-2673: Attempting to stop 'ora.storage' on 'host03'
CRS-2677: Stop of 'ora.storage' on 'host03' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host03'
CRS-2677: Stop of 'ora.FRA.dg' on 'host02' succeeded
CRS-2677: Stop of 'ora.DATA.dg' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host02'
CRS-2677: Stop of 'ora.asm' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.ASMNET1LSNR_ASM.lsnr' on 'host02'
CRS-2677: Stop of 'ora.ctssd' on 'host03' succeeded
CRS-2677: Stop of 'ora.evmd' on 'host03' succeeded
```

```
CRS-2677: Stop of 'ora.ASMNET1LSNR_ASM.lsnr' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.ons' on 'host02'
CRS-2677: Stop of 'ora.ons' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.net1.network' on 'host02'
CRS-2677: Stop of 'ora.net1.network' on 'host02' succeeded
CRS-2792: Shutdown of Cluster Ready Services-managed resources on
'host02' has completed
CRS-2677: Stop of 'ora.crsd' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.ctssd' on 'host02'
CRS-2673: Attempting to stop 'ora.evmd' on 'host02'
CRS-2673: Attempting to stop 'ora.storage' on 'host02'
CRS-2677: Stop of 'ora.storage' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host02'
CRS-2677: Stop of 'ora.evmd' on 'host02' succeeded
CRS-2677: Stop of 'ora.asm' on 'host03' succeeded
CRS-2673: Attempting to stop 'ora.cluster_interconnect.haip' on
'host03'
CRS-2677: Stop of 'ora.ctssd' on 'host02' succeeded
CRS-2677: Stop of 'ora.cluster_interconnect.haip' on 'host03'
succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host03'
CRS-2677: Stop of 'ora.cssd' on 'host03' succeeded
CRS-2677: Stop of 'ora.asm' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.cluster_interconnect.haip' on
'host02'
CRS-2677: Stop of 'ora.cluster_interconnect.haip' on 'host02'
succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host02'
CRS-2677: Stop of 'ora.cssd' on 'host02' succeeded
[root@host01 ~]#
```

7. Restart Clusterware on all nodes.

```
[root@host01 ~]# crsctl start cluster -all
CRS-2672: Attempting to start 'ora.evmd' on 'host04'
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'host04'
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'host05'
CRS-2672: Attempting to start 'ora.evmd' on 'host05'
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'host02'
CRS-2672: Attempting to start 'ora.evmd' on 'host02'
CRS-2676: Start of 'ora.evmd' on 'host04' succeeded
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'host01'
CRS-2672: Attempting to start 'ora.evmd' on 'host01'
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'host03'
CRS-2672: Attempting to start 'ora.evmd' on 'host03'
CRS-2676: Start of 'ora.cssdmonitor' on 'host04' succeeded
CRS-2672: Attempting to start 'ora.cssd' on 'host04'
CRS-2672: Attempting to start 'ora.diskmon' on 'host04'
CRS-2676: Start of 'ora.evmd' on 'host05' succeeded
CRS-2676: Start of 'ora.diskmon' on 'host04' succeeded
CRS-2676: Start of 'ora.evmd' on 'host02' succeeded
CRS-2676: Start of 'ora.cssdmonitor' on 'host05' succeeded
```

```
CRS-2672: Attempting to start 'ora.cssd' on 'host05'
CRS-2676: Start of 'ora.evmd' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.diskmon' on 'host05'
CRS-2676: Start of 'ora.cssdmonitor' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.cssd' on 'host03'
CRS-2672: Attempting to start 'ora.diskmon' on 'host03'
CRS-2676: Start of 'ora.cssdmonitor' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.cssd' on 'host02'
CRS-2672: Attempting to start 'ora.diskmon' on 'host02'
CRS-2676: Start of 'ora.diskmon' on 'host02' succeeded
CRS-2676: Start of 'ora.diskmon' on 'host05' succeeded
CRS-2676: Start of 'ora.cssdmonitor' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.cssd' on 'host01'
CRS-2672: Attempting to start 'ora.diskmon' on 'host01'
CRS-2676: Start of 'ora.diskmon' on 'host01' succeeded
CRS-2676: Start of 'ora.evmd' on 'host03' succeeded
CRS-2676: Start of 'ora.diskmon' on 'host03' succeeded
CRS-2676: Start of 'ora.cssd' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.ctssd' on 'host02'
CRS-2672: Attempting to start 'ora.cluster_interconnect.haip' on
'host02'
CRS-2676: Start of 'ora.cssd' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.ctssd' on 'host01'
CRS-2672: Attempting to start 'ora.cluster_interconnect.haip' on
'host01'
CRS-2676: Start of 'ora.cssd' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.ctssd' on 'host03'
CRS-2672: Attempting to start 'ora.cluster_interconnect.haip' on
'host03'
CRS-2676: Start of 'ora.ctssd' on 'host02' succeeded
CRS-2676: Start of 'ora.ctssd' on 'host01' succeeded
CRS-2676: Start of 'ora.ctssd' on 'host03' succeeded
CRS-2676: Start of 'ora.cluster_interconnect.haip' on 'host02'
succeeded
CRS-2672: Attempting to start 'ora.asm' on 'host02'
CRS-2676: Start of 'ora.cluster_interconnect.haip' on 'host01'
succeeded
CRS-2672: Attempting to start 'ora.asm' on 'host01'
CRS-2676: Start of 'ora.asm' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.storage' on 'host01'
CRS-2676: Start of 'ora.cluster_interconnect.haip' on 'host03'
succeeded
CRS-2672: Attempting to start 'ora.asm' on 'host03'
CRS-2676: Start of 'ora.asm' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.storage' on 'host03'
CRS-2676: Start of 'ora.asm' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.storage' on 'host02'
CRS-2676: Start of 'ora.storage' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'host02'
CRS-2676: Start of 'ora.crsd' on 'host02' succeeded
CRS-2676: Start of 'ora.cssd' on 'host05' succeeded
CRS-2672: Attempting to start 'ora.storage' on 'host05'
```

```
CRS-2672: Attempting to start 'ora.ctssd' on 'host05'
CRS-2672: Attempting to start 'ora.cluster_interconnect.haip' on
'host05'
CRS-2676: Start of 'ora.storage' on 'host05' succeeded
CRS-2676: Start of 'ora.cluster_interconnect.haip' on 'host05'
succeeded
CRS-2676: Start of 'ora.cssd' on 'host04' succeeded
CRS-2672: Attempting to start 'ora.storage' on 'host04'
CRS-2672: Attempting to start 'ora.ctssd' on 'host04'
CRS-2672: Attempting to start 'ora.cluster_interconnect.haip' on
'host04'
CRS-2676: Start of 'ora.storage' on 'host04' succeeded
CRS-2676: Start of 'ora.cluster_interconnect.haip' on 'host04'
succeeded
CRS-2676: Start of 'ora.ctssd' on 'host05' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'host05'
CRS-2676: Start of 'ora.ctssd' on 'host04' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'host04'
CRS-2676: Start of 'ora.crsd' on 'host04' succeeded
CRS-2676: Start of 'ora.crsd' on 'host05' succeeded
CRS-2676: Start of 'ora.storage' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'host03'
CRS-2676: Start of 'ora.crsd' on 'host03' succeeded
CRS-2676: Start of 'ora.storage' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'host01'
CRS-2676: Start of 'ora.crsd' on 'host01' succeeded
[root@host01 ~]#
```

8. Use `ifconfig` to determine view how Clusterware has implemented HAIP.

```
[root@host01 ~]# ifconfig -a
eth0 Link encap:Ethernet  HWaddr 00:16:3E:00:01:01
     inet addr:192.0.2.101  Bcast:192.0.2.255  Mask:255.255.255.0
     inet6 addr: fe80::216:3eff:fe00:101/64 Scope:Link
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:5901334 errors:0 dropped:0 overruns:0 frame:0
     TX packets:5936201 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:1000
     RX bytes:12336082590 (11.4 GiB) TX bytes:48768508465 (45.4 GiB)
     Interrupt:33

eth0:1 Link encap:Ethernet  HWaddr 00:16:3E:00:01:01
       inet addr:192.0.2.252  Bcast:192.0.2.255  Mask:255.255.255.0
       UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
       Interrupt:33

...

eth1 Link encap:Ethernet  HWaddr 00:16:3E:00:01:11
     inet addr:192.168.1.101 Bcast:192.168.1.255  Mask:255.255.255.0
     inet6 addr: fe80::216:3eff:fe00:111/64 Scope:Link
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```
          RX packets:95575544 errors:0 dropped:0 overruns:0 frame:0
          TX packets:96277357 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:53922515949 (50.2 GiB) TX bytes:61819202855 (57.5 GiB)
          Interrupt:34

eth1:1 Link encap:Ethernet  HWaddr 00:16:3E:00:01:11
          inet addr:169.254.34.33  Bcast:169.254.127.255
          Mask:255.255.128.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:34

eth2 Link encap:Ethernet  HWaddr 00:16:3E:00:01:21
          inet addr:192.168.2.101 Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::216:3eff:fe00:121/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:538313 errors:0 dropped:0 overruns:0 frame:0
          TX packets:321654 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:186776449 (178.1 MiB)  TX bytes:174451936 (166.3 MiB)
          Interrupt:35

eth2:1 Link encap:Ethernet  HWaddr 00:16:3E:00:01:21
          inet addr:169.254.207.210 Bcast:169.254.255.255
          Mask:255.255.128.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:35

lo   Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:10695365 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10695365 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:31305491786 (29.1 GiB) TX bytes:31305491786 (29.1 GiB)

[root@host01 ~]#
```

Instead of using `eth1` and `eth2` directly, Clusterware has configured private VIPs on `eth1:1` and `eth2:1` using IP addresses on the reserved `169.254.*.*` subnet to support HAIP.

9. Close all terminal windows opened for these practices.

Practices for Lesson 8: Policy-Based Cluster Management

Chapter 8

Practices for Lesson 8: Overview

Practices Overview

In this practice you will configure server categories and the policy set. You will examine the effect of various changes to verify the dynamic nature of policy-based cluster management. Finally you will see how easy it is to activate policies.

Practice 8-1: Configuring and Using Policy-Based Cluster Management

Overview

In this practice you will configure server categories and the policy set. You will examine the effect of various changes to verify the dynamic nature of policy-based cluster management.

1. Connect to the first node of your cluster as the `grid` user. You can use the `oraenv` script to set your environment correctly*. **Do not use the `root` account at any time during this practice!***

```
[vncuser@classroom_pc]# ssh grid@host01
grid@host01's password:
Last login: Mon May 20 20:01:43 2013 from dns.example.com
[grid@host01 ~]$
[grid@host01~] $ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base has been set to /u01/app/grid
```

2. Examine the extended server attributes associated with `host01, host02` and `host03`. Notice the amount of physical memory associated with each server (3806MB + for `host01`, 3128MB + for `host02` and `host03`). Notice also that the attributes identify these nodes as Hub Nodes in the cluster.

```
[grid@host01 ~]$ crsctl status server host01 -f
NAME=host01
MEMORY_SIZE=3806
CPU_COUNT=1
CPU_CLOCK_RATE=2
CPU_HYPERTHREADING=0
CPU_EQUIVALENCY=1000
DEPLOYMENT=other
CONFIGURED_CSS_ROLE=hub
RESOURCE_USE_ENABLED=1
SERVER_LABEL=
PHYSICAL_HOSTNAME=
STATE=ONLINE
ACTIVE_POOLS=ora.orcldb
STATE_DETAILS=AUTOSTARTING RESOURCES
ACTIVE_CSS_ROLE=hub

[grid@host01 ~]$ crsctl status server host02 -f
NAME=host02
MEMORY_SIZE=3142
CPU_COUNT=2
CPU_CLOCK_RATE=5844
CPU_HYPERTHREADING=0
CPU_EQUIVALENCY=1000
DEPLOYMENT=other
CONFIGURED_CSS_ROLE=hub
RESOURCE_USE_ENABLED=1
```

```
SERVER_LABEL=
PHYSICAL_HOSTNAME=
STATE=ONLINE
ACTIVE_POOLS=ora.orcldb
STATE_DETAILS=AUTOSTARTING RESOURCES
ACTIVE_CSS_ROLE=hub

[grid@host01 ~]$ crsctl status server host03 -f
MEMORY_SIZE=3142
CPU_COUNT=2
CPU_CLOCK_RATE=2
CPU_HYPERTHREADING=0
CPU_EQUIVALENCY=1000
DEPLOYMENT=other
CONFIGURED_CSS_ROLE=hub
RESOURCE_USE_ENABLED=1
SERVER_LABEL=
PHYSICAL_HOSTNAME=
STATE=ONLINE
ACTIVE_POOLS=ora.orcldb
STATE_DETAILS=AUTOSTARTING RESOURCES
ACTIVE_CSS_ROLE=hub

[grid@host01 ~]$
```

3. Examine the extended server attributes associated with `host04` and `host05`. Notice that these nodes contain less physical memory on each node (1557MB) and that they are identified as Leaf Nodes in this Flex Cluster.

```
[grid@host01 ~]$ crsctl status server host04 -f
NAME=host04
MEMORY_SIZE=1557
CPU_COUNT=1
CPU_CLOCK_RATE=2
CPU_HYPERTHREADING=0
CPU_EQUIVALENCY=1000
DEPLOYMENT=other
CONFIGURED_CSS_ROLE=leaf
RESOURCE_USE_ENABLED=1
SERVER_LABEL=
PHYSICAL_HOSTNAME=
STATE=ONLINE
ACTIVE_POOLS=Free
STATE_DETAILS=
ACTIVE_CSS_ROLE=leaf

[grid@host01 ~]$ crsctl status server host05 -f
NAME=host05
MEMORY_SIZE=1557
CPU_COUNT=1
CPU_CLOCK_RATE=2
CPU_HYPERTHREADING=0
CPU_EQUIVALENCY=1000
```

```
DEPLOYMENT=other
CONFIGURED_CSS_ROLE=leaf
RESOURCE_USE_ENABLED=1
SERVER_LABEL=
PHYSICAL_HOSTNAME=
STATE=ONLINE
ACTIVE_POOLS=Free
STATE_DETAILS=
ACTIVE_CSS_ROLE=leaf

 [grid@host01 ~]$
```

4. Examine the built-in category definitions. These are implicitly used to categorize the cluster nodes as Hub Nodes or Leaf Nodes based on the `ACTIVE_CSS_ROLE` setting. Note that these categories also exist in a standard cluster, however in a standard cluster all the nodes are designated as Hub Nodes.

```
 [grid@host01 ~]$ crsctl status category
NAME=ora.hub.category
ACL=owner:root:rwx,pgrp:root:r-x,other::r--
ACTIVE_CSS_ROLE=hub
EXPRESSION=

NAME=ora.leaf.category
ACL=owner:root:rwx,pgrp:root:r-x,other::r--
ACTIVE_CSS_ROLE=leaf
EXPRESSION=

 [grid@host01 ~]$
```

5. Examine the servers that are associated with the inbuilt categories. Confirm that the server-to-category mappings are as expected.

```
 [grid@host01 ~]$ crsctl status server -category ora.hub.category
NAME=host01
STATE=ONLINE

NAME=host02
STATE=ONLINE

NAME=host03
STATE=ONLINE

 [grid@host01 ~]$ crsctl status server -category ora.leaf.category
NAME=host04
STATE=ONLINE

NAME=host05
STATE=ONLINE

 [grid@host01 ~]$
```

6. Create a user defined category which includes the servers that contain more than 3000MB of system memory.

```
[grid@host01 ~]$ crsctl add category big -attr
"EXPRESSION='(MEMORY_SIZE > 3000)'"
[grid@host01 ~]$
```

7. Examine the category definition that you created in the previous step. Notice that the category definition includes `ACTIVE_CSS_ROLE=hub` by default. You could modify the `ACTIVE_CSS_ROLE` attribute if you wanted to categorize Leaf Nodes.

```
[[grid@host01 ~]$ crsctl status category big
NAME=big
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r--
ACTIVE_CSS_ROLE=hub
EXPRESSION=(MEMORY_SIZE > 3000)
```

8. Examine the servers associated with the big category. Confirm that `host01`, `host02` and `host03` are associated with the big category.

```
[grid@host01 ~]$ crsctl status server -category big
NAME=host01
STATE=ONLINE

NAME=host02
STATE=ONLINE

NAME=host03
STATE=ONLINE

 [grid@host01 ~]$
```

9. Examine the categories associated with the `host01` server. Notice that a server can be associated with multiple categories.

```
[grid@host01 ~]$ crsctl status category -server host01
NAME=big
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r--
ACTIVE_CSS_ROLE=hub
EXPRESSION=(MEMORY_SIZE > 3000)

NAME=ora.hub.category
ACL=owner:root:rwx,pgrp:root:r-x,other::r--
ACTIVE_CSS_ROLE=hub
EXPRESSION=

 [grid@host01 ~]$
```

At this point you have configured a category and examined category definitions along with server-to-category associations. Next, you will make use of the category definitions as you configure and exercise policy-based cluster management.

10. Examine the policy set. At this point, no policy set configuration has been performed so the only policy listed is the `Current` policy. The `Current` policy is a special built-in policy which contains all of the currently active server pool definitions.

```
[grid@host01 ~]$ crsctl status policyset
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r--
LAST_ACTIVATED_POLICY=
SERVER_POOL_NAMES=Free
POLICY
  NAME=Current
  DESCRIPTION=This policy is built-in and managed automatically to
reflect current configuration
  SERVERPOOL
    NAME=Free
    IMPORTANCE=0
    MAX_SIZE=-1
    MIN_SIZE=0
    SERVER_CATEGORY=
    SERVER_NAMES=
  SERVERPOOL
    NAME=Generic
    IMPORTANCE=0
    MAX_SIZE=-1
    MIN_SIZE=0
    SERVER_CATEGORY=
    SERVER_NAMES=
  SERVERPOOL
    NAME=ora.orcldb
    IMPORTANCE=0
    MAX_SIZE=3
    MIN_SIZE=0
    SERVER_CATEGORY=ora.hub.category
    SERVER_NAMES=
[grid@host01 ~]$
```

11. Examine the policy set definition provided in the file located at `/stage/GRID/labs/less_08/policyset.txt`. The policy set definition contains two policies. The day policy enables the `orcldb` server pool to use two Hub Nodes while not providing any allocation for the `bigpool` server pool. The night policy still prioritizes the `orcldb` server pool, however it also provides an allocation for the `bigpool` server pool. Notice that the `bigpool` server pool references the big server category that you defined earlier in this practice. Notice also that the policy set does not list the `my_app_pool` server pool in the `SERVER_POOL_NAMES` attribute. This demonstrates how it is possible for you to define server pools that are outside policy set management.

```
[grid@host01 ~]$ cat /stage/GRID/labs/less_08/policyset.txt
SERVER_POOL_NAMES=Free bigpool ora.orcldb
POLICY
  NAME=day
  DESCRIPTION=The day policy
  SERVERPOOL
    NAME=ora.orcldb
    IMPORTANCE=10
    MAX_SIZE=3
    MIN_SIZE=1
    SERVER_CATEGORY=ora.hub.category
  SERVERPOOL
    NAME=bigpool
    IMPORTANCE=0
    MAX_SIZE=0
    MIN_SIZE=0
    SERVER_CATEGORY=big
POLICY
  NAME=night
  DESCRIPTION=The night policy
  SERVERPOOL
    NAME=ora.orcldb
    IMPORTANCE=10
    MAX_SIZE=3
    MIN_SIZE=1
    SERVER_CATEGORY=ora.hub.category
  SERVERPOOL
    NAME=bigpool
    IMPORTANCE=5
    MAX_SIZE=1
    MIN_SIZE=1
    SERVER_CATEGORY=big

[grid@host01 ~]$
```

12. Modify the policy set to load the configuration file.

```
[grid@host01 ~]$ crsctl modify policyset -file
/stage/GRID/labs/less_08/policyset.txt
[grid@host01 ~]$
```

13. Reexamine the policy set to confirm that the configuration file was loaded in the previous step. Notice that the `LAST_ACTIVATED_POLICY` attribute is not set and the `Current` policy is unchanged from what you observed earlier. This is because no policies have been activated yet.

```
[grid@host01 ~]$ crsctl status policyset
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r--
LAST_ACTIVATED_POLICY=
SERVER_POOL_NAMES=Free bigpool ora.orcldb
POLICY
  NAME=Current
  DESCRIPTION=This policy is built-in and managed automatically to
reflect current configuration
  SERVERPOOL
    NAME=Free
    IMPORTANCE=0
    MAX_SIZE=-1
    MIN_SIZE=0
    SERVER_CATEGORY=
    SERVER_NAMES=
  SERVERPOOL
    NAME=Generic
    IMPORTANCE=0
    MAX_SIZE=-1
    MIN_SIZE=0
    SERVER_CATEGORY=
    SERVER_NAMES=
  SERVERPOOL
    NAME=ora.orcldb
    IMPORTANCE=0
    MAX_SIZE=3
    MIN_SIZE=0
    SERVER_CATEGORY=ora.hub.category
    SERVER_NAMES=
POLICY
  NAME=day
  DESCRIPTION=The day policy
  SERVERPOOL
    NAME=Free
    IMPORTANCE=0
    MAX_SIZE=-1
    MIN_SIZE=0
    SERVER_CATEGORY=
    SERVER_NAMES=
  SERVERPOOL
    NAME=bigpool
    IMPORTANCE=0
    MAX_SIZE=0
    MIN_SIZE=0
    SERVER_CATEGORY=big
    SERVER_NAMES=
  SERVERPOOL
    NAME=ora.orcldb
    IMPORTANCE=10
    MAX_SIZE=2
    MIN_SIZE=1
    SERVER_CATEGORY=ora.hub.category
    SERVER_NAMES=
POLICY
  NAME=night
  DESCRIPTION=The night policy
  SERVERPOOL
    NAME=Free
    IMPORTANCE=0
    MAX_SIZE=-1
    MIN_SIZE=0
    SERVER_CATEGORY=
    SERVER_NAMES=
  SERVERPOOL
    NAME=bigpool
    IMPORTANCE=5
    MAX_SIZE=1
    MIN_SIZE=1
    SERVER_CATEGORY=big
    SERVER_NAMES=
  SERVERPOOL
    NAME=ora.orcldb
    IMPORTANCE=10
    MAX_SIZE=3
    MIN_SIZE=1
    SERVER_CATEGORY=ora.hub.category
    SERVER_NAMES=
[grid@host01 ~]$
```

14. Activate the day policy.

```
[grid@host01 ~]$ crsctl modify policyset -attr
"LAST_ACTIVATED_POLICY='day'"
[grid@host01 ~]$
```

15. Reexamine the policy set. Confirm that `LAST_ACTIVATED_POLICY=day` and that the `Current` policy settings reflect the day policy.

```
[grid@host01 ~]$ crsctl status policyset
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r--
LAST_ACTIVATED_POLICY=day
SERVER_POOL_NAMES=Free bigpool ora.orcldb
POLICY
  NAME=Current
  DESCRIPTION=This policy is built-in and managed automatically to
reflect current configuration
  SERVERPOOL
    NAME=Free
    IMPORTANCE=0
    MAX_SIZE=-1
    MIN_SIZE=0
    SERVER_CATEGORY=
    SERVER_NAMES=
  SERVERPOOL
    NAME=Generic
    IMPORTANCE=0
    MAX_SIZE=-1
    MIN_SIZE=0
    SERVER_CATEGORY=
    SERVER_NAMES=
  SERVERPOOL
    NAME=bigpool
    IMPORTANCE=0
    MAX_SIZE=0
    MIN_SIZE=0
    SERVER_CATEGORY=big
    SERVER_NAMES=
  SERVERPOOL
    NAME=ora.orcldb
    IMPORTANCE=10
    MAX_SIZE=2
    MIN_SIZE=1
    SERVER_CATEGORY=ora.hub.category
    SERVER_NAMES=
POLICY
  NAME=day
  DESCRIPTION=The day policy
  SERVERPOOL
    NAME=Free
    IMPORTANCE=0
    MAX_SIZE=-1
    MIN_SIZE=0
    SERVER_CATEGORY=
    SERVER_NAMES=
  SERVERPOOL
    NAME=bigpool
    IMPORTANCE=0
    MAX_SIZE=0
    MIN_SIZE=0
    SERVER_CATEGORY=big
    SERVER_NAMES=
  SERVERPOOL
    NAME=ora.orcldb
    IMPORTANCE=10
    MAX_SIZE=2
    MIN_SIZE=1
    SERVER_CATEGORY=ora.hub.category
    SERVER_NAMES=
POLICY
  NAME=night
  DESCRIPTION=The night policy
  SERVERPOOL
    NAME=Free
    IMPORTANCE=0
    MAX_SIZE=-1
    MIN_SIZE=0
    SERVER_CATEGORY=
    SERVER_NAMES=
  SERVERPOOL
    NAME=bigpool
    IMPORTANCE=5
    MAX_SIZE=1
    MIN_SIZE=1
    SERVER_CATEGORY=big
    SERVER_NAMES=
  SERVERPOOL
    NAME=ora.orcldb
    IMPORTANCE=10
    MAX_SIZE=3
    MIN_SIZE=1
    SERVER_CATEGORY=ora.hub.category
    SERVER_NAMES=
[grid@host01 ~]$
```

16. Examine the server pool allocations. Confirm the Hub Nodes are still allocated to the `orcldb` server pool in line with the day policy.

```
[grid@host01 ~]$ crsctl status serverpool
NAME=Free
ACTIVE_SERVERS=host04 host05

NAME=Generic
ACTIVE_SERVERS=

NAME=bigpool
ACTIVE_SERVERS=

NAME=ora.orcldb
ACTIVE_SERVERS=host01 host02 host03

[grid@host01 ~]$
```

17. Activate the `night` policy.

```
[grid@host01 ~]$ crsctl modify policyset -attr
"LAST_ACTIVATED_POLICY='night'"

[grid@host01 ~]$
```

18. Examine the server pool allocations. Confirm that the servers allocated to each pool are consistent with the night policy.

```
[grid@host01 ~]$ crsctl status serverpool
NAME=Free
ACTIVE_SERVERS=host04 host05

NAME=Generic
ACTIVE_SERVERS=

NAME=bigpool
ACTIVE_SERVERS=host01

NAME=ora.orcldb
ACTIVE_SERVERS=host02 host03

[grid@host01 ~]$
```

19. One of the side-effects of activating the night policy is that the RAC database (`orcl`) that previously ran on both `host01`, `host02`, and `host03` now only runs on two nodes. Confirm that only two instances of the `orcl` database are running. This demonstrates how Oracle Clusterware automatically starts and stops required resources when a policy change is made.

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

20. To illustrate the dynamic nature of policy-based cluster management, modify the `big` category so that no servers can be associated with it. Notice that the category change immediately causes a server re-allocation, which in turn causes an instance of the `orcl` database to start up.

```
[grid@host01 ~]$ crsctl modify category big -attr
"EXPRESSION='(MEMORY_SIZE > 8000)'"
CRS-2672: Attempting to start 'ora.orcl.db' on 'host01'
CRS-2676: Start of 'ora.orcl.db' on 'host01' succeeded
[grid@host01 ~]$
```

21. Reexamine the server pool allocations. Notice that no servers are associated with the `bigpool` server pool because of the change to the big category, which in turn results in three servers being allocated to the `orcldb` server pool. Confirm that `ora.orcl.db` is now placed on three servers.

```
[grid@host01 ~]$ crsctl status serverpool
NAME=Free
ACTIVE_SERVERS=host04 host05

NAME=Generic
ACTIVE_SERVERS=

NAME=bigpool
ACTIVE_SERVERS=

NAME=ora.orcldb
ACTIVE_SERVERS=host01 host02 host03

[grid@host01 ~]$ srvctl status database -d 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
[grid@host01 ~]$
```

22. Close all terminal windows opened for this practice.

Practices for Lesson 9: Upgrading and Patching Grid Infrastructure

Chapter 9

Practices for Lesson 9: Overview

Practices Overview

There are no practices for this lesson.

Practices for Lesson 10: Troubleshooting Oracle Clusterware

Chapter 10

Practices for Lesson 10: Overview

Practices Overview

In this practice, you will work with Oracle Clusterware log files and learn to use the `ocrdump` and `cluvfy` utilities.

Practice 10-1: Working with Log Files

Overview

In this practice, you will examine the Oracle Clusterware alert log and then package various log files into an archive format suitable to send to My Oracle Support.

1. Connect to host01 as the grid user, locate and view the contents of the Oracle Clusterware alert log.

```
[vncuser@classroom_pc ~]$ ssh grid@host01
grid@host01's password:
Last login: Wed May 22 15:37:18 2013 from dns.example.com
[grid@host01 ~]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base has been set to /u01/app/grid
[grid@host01 ~]$
[grid@host01 ~]$ cd /u01/app/12.1.0/grid/log/host01


[grid@host01 host01]$ view alerthost01.log

2013-05-07 20:13:49.505:
[client(18069)]CRS-2101:The OLR was formatted using version 4.
2013-05-07 20:14:43.563:
[ohasd(18717)]CRS-0714:Oracle Clusterware Release 12.1.0.1.0 -
Production Copyright 1996, 2010 Oracle. All rights reserved.
2013-05-07 20:14:43.656:
[ohasd(18717)]CRS-2112:The OLR service started on node host01.
2013-05-07 20:14:43.783:
[ohasd(18717)]CRS-1301:Oracle High Availability Service started on
node host01.
[client(18925)]CRS-10001:07-May-13 20:14 ACFS-9200: Supported
2013-05-07 20:15:15.180:
[ohasd(19083)]CRS-0714:Oracle Clusterware Release 12.1.0.1.0 -
Production Copyright 1996, 2010 Oracle. All rights reserved.
2013-05-07 20:15:15.249:
[ohasd(19083)]CRS-2112:The OLR service started on node host01.
2013-05-07 20:15:15.263:
[ohasd(19083)]CRS-1301:Oracle High Availability Service started on
node host01.
[client(19213)]CRS-10001:07-May-13 20:15 ACFS-9300: ADVM/ACFS
distribution files found.
[client(19219)]CRS-10001:07-May-13 20:15 ACFS-9312: Existing
ADVM/ACFS installation detected.
[client(19221)]CRS-10001:07-May-13 20:15 ACFS-9314: Removing
previous ADVM/ACFS installation.
[client(19228)]CRS-10001:07-May-13 20:15 ACFS-9315: Previous
ADVM/ACFS components successfully removed.
[client(19230)]CRS-10001:07-May-13 20:15 ACFS-9307: Installing
requested ADVM/ACFS software.
[client(19263)]CRS-10001:07-May-13 20:15 ACFS-9308: Loading
installed ADVM/ACFS drivers.
```

```
[client(19265)]CRS-10001:07-May-13 20:15 ACFS-9321: Creating udev
for ADVM/ACFS.
[client(19267)]CRS-10001:07-May-13 20:15 ACFS-9323: Creating module
dependencies - this may take some time.
[client(29783)]CRS-10001:07-May-13 20:16 ACFS-9154: Loading
'oracleoks.ko' driver.
[client(29793)]CRS-10001:07-May-13 20:16 ACFS-9154: Loading
'oracleadvm.ko' driver.
[client(29827)]CRS-10001:07-May-13 20:16 ACFS-9154: Loading
'oracleacfs.ko' driver.
[client(29882)]CRS-10001:07-May-13 20:16 ACFS-9327: Verifying
ADVM/ACFS devices.
[client(29884)]CRS-10001:07-May-13 20:16 ACFS-9156: Detecting
control device '/dev/asm/.asm_ctl_spec'.
[client(29888)]CRS-10001:07-May-13 20:16 ACFS-9156: Detecting
control device '/dev/ofsctl'.
[client(29893)]CRS-10001:07-May-13 20:16 ACFS-9309: ADVM/ACFS
installation correctness verified.
2013-05-07 20:16:43.548:
[ohasd(29967)]CRS-0714:Oracle Clusterware Release 12.1.0.1.0 -
Production Copyright 1996, 2010 Oracle. All rights reserved.
2013-05-07 20:16:43.606:
[ohasd(29967)]CRS-2112:The OLR service started on node host01.
2013-05-07 20:16:43.621:
[ohasd(29967)]CRS-1301:Oracle High Availability Service started on
node host01.
2013-05-07 20:16:51.014:
[gpnpd(30096)]CRS-2328:GPNPD started on node host01.
2013-05-07 20:16:53.490:
"alerthost01.log" [readonly] 16919L, 662914C
```

2. Navigate to the Oracle Cluster Synchronization Services daemon log directory and determine whether any log archives exist.

```
[grid@host01 host01]$ cd cssd
[grid@host01 cssd]$ ls -la ocssd*
-rw-r--r-- 1 grid oinstall 52635156 May 28 02:24 ocssd.l01
-rw-r--r-- 1 grid oinstall 52634868 May 27 11:30 ocssd.l02
-rw-r--r-- 1 grid oinstall 52634921 May 26 20:43 ocssd.l03
-rw-r--r-- 1 grid oinstall 52634750 May 26 05:49 ocssd.l04
-rw-r--r-- 1 grid oinstall 52634063 May 25 15:03 ocssd.l05
-rw-r--r-- 1 grid oinstall 52629436 May 25 00:09 ocssd.l06
-rw-r--r-- 1 grid oinstall 52630342 May 24 09:28 ocssd.l07
-rw-r--r-- 1 grid oinstall 52631859 May 23 19:06 ocssd.l08
-rw-r--r-- 1 grid oinstall 52631497 May 23 04:28 ocssd.l09
-rw-r--r-- 1 grid oinstall 52634557 May 22 13:38 ocssd.l10
-rw-r--r-- 1 grid oinstall 30723966 May 28 11:00 ocssd.log
[grid@host01 cssd]$
```

3. Switch to the `root` user and set up the environment variables for the Grid Infrastructure. Change to the `ORACLE_HOME/bin` directory and run the `diagcollection.pl` script to gather all log files that can be sent to My Oracle Support for problem analysis.

```
[grid@host01 cssd]$ su -
Password:

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

[root@host01 ~]# cd $ORACLE_HOME/bin

[root@host01 bin]# ./diagcollection.pl --collect $ORACLE_HOME

Production Copyright 2004, 2010, Oracle.  All rights reserved
Cluster Ready Services (CRS) diagnostic collection tool
The following CRS diagnostic archives will be created in the local
directory.
crsData_host01_20130919_1022.tar.gz -> logs,traces and cores from
CRS home. Note: core files will be packaged only with the --core
option.
ocrData_host01_20130919_1022.tar.gz -> ocrdump, ocrcheck etc
coreData_host01_20130919_1022.tar.gz -> contents of CRS core files
in text format

osData_host01_20130919_1022.tar.gz -> logs from Operating System
Collecting crs data
/bin/tar:
log/host01/agent/crsd/orarootagent_root/orarootagent_root.log: file
changed as we read it
/bin/tar: log/host01/cssd/ocssd.log: file changed as we read it
Collecting OCR data
Collecting information from core files
No corefiles found
The following diagnostic archives will be created in the local
directory.
acfsData_host01_20130919_1022.tar.gz -> logs from acfs log.
Collecting acfs data
Collecting OS logs
Collecting sysconfig data
[root@host01 bin]#
```

4. List the resulting log file archives that were generated with the `diagcollection.pl` script.

```
[root@host01 bin]# ls -la *.gz
-rw-r--r-- 1 root root     1881 Sep 19 10:23
acfsData_host01_20130919_1022.tar.gz
-rw-r--r-- 1 root root 11929110 Sep 19 10:22
crsData_host01_20130919_1022.tar.gz
-rw-r--r-- 1 root root    48438 Sep 19 10:23
ocrData_host01_20130919_1022.tar.gz
```

```
-rw-r--r-- 1 root root   119140 Sep 19 10:23
osData_host01_20130919_1022.tar.gz
[root@host01 bin]#
```

5. Exit the root account and return to the `grid` account.

```
[root@host01 bin]# exit
logout
[grid@host01 cssd]$
```

Practice 10-2: Working with OCRDUMP

Overview

In this practice, you will work with the OCRDUMP utility and dump the binary file into both text and XML representations.

1. While connected to the grid account, dump the contents of the OCR to the standard output and count the number of lines of output.

```
[grid@host01 ~]$ ocrdump -stdout | wc -l
1352
[grid@host01 ~]$
```

2. Switch to the `root` user, dump the contents of the OCR to the standard output and count the number of lines of output. Compare your results with the previous step. How do the results differ?

```
[grid@host01 ~]$ su -
Password:

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

[root@host01 ~]# ocrdump -stdout | wc -l
5435
[root@host01 ~]#
```

3. Dump the first 25 lines of the OCR to standard output using XML format.

```
[root@host01 ~]# ocrdump -stdout -xml | head -25
<OCRDUMP>

<TIMESTAMP>05/24/2013 19:16:14</TIMESTAMP>
<COMMAND>/u01/app/12.1.0/grid/bin/ocrdump.bin -stdout -xml
</COMMAND>

<KEY>
<NAME>SYSTEM</NAME>
<VALUE_TYPE>UNDEF</VALUE_TYPE>
<VALUE><![CDATA[]]></VALUE>
<USER_PERMISSION>PROCR_ALL_ACCESS</USER_PERMISSION>
<GROUP_PERMISSION>PROCR_READ</GROUP_PERMISSION>
<OTHER_PERMISSION>PROCR_READ</OTHER_PERMISSION>
<USER_NAME>root</USER_NAME>
<GROUP_NAME>root</GROUP_NAME>

<KEY>
<NAME>SYSTEM.version</NAME>
<VALUE_TYPE>UB4 (10)</VALUE_TYPE>
<VALUE><![CDATA[5]]></VALUE>
```

```
<USER_PERMISSION>PROCR_ALL_ACCESS</USER_PERMISSION>
<GROUP_PERMISSION>PROCR_READ</GROUP_PERMISSION>
<OTHER_PERMISSION>PROCR_READ</OTHER_PERMISSION>
<USER_NAME>root</USER_NAME>
<GROUP_NAME>root</GROUP_NAME>

[root@host01 ~]#4
```

4. Create an XML file dump of the OCR in the `/tmp` directory. Name the dump file `ocr_current_dump.xml`.

```
[root@host01 ~]# ocrdump -xml /tmp/ocr_current_dump.xml
[root@host01 ~]#
```

5. Find the node and the directory that contains the automatic backup of the OCR from 24 hours ago.

```
[root@host01 ~]# ocrconfig -showbackup

host02     2013/05/24 19:13:46
/u01/app/12.1.0/grid/cdata/cluster01/backup00.ocr

host02     2013/05/24 15:13:07
/u01/app/12.1.0/grid/cdata/cluster01/backup01.ocr

host02     2013/05/24 11:13:07
/u01/app/12.1.0/grid/cdata/cluster01/backup02.ocr

host02     2013/05/23 07:12:59
/u01/app/12.1.0/grid/cdata/cluster01/day.ocr
...
[root@host01 ~]#
```

6. Copy the 24-hour-old automatic backup of the OCR into the `/tmp` directory. This is not a dump, but rather an actual backup of the OCR. (If the daily backup of the OCR is not there, use the oldest backup on the list.)

   Note: It may be necessary to use `scp` if the file is located on a different node.

```
[root@host01 ~]# scp
host02:/u01/app/12.1.0/grid/cdata/cluster01/day.ocr /tmp/day.ocr
root@host02's password:
day.ocr
100% 5764KB   1.9MB/s   00:03
[root@host01 ~]#
```

7. Dump the contents of the day.ocr backup OCR file in the XML format saving the file in the `/tmp directory`. Name the file as `day_ocr.xml`.

```
[root@host01 ~]# ocrdump -xml -backupfile /tmp/day.ocr
/tmp/day_ocr.xml
[root@host01 ~]#
-rw-r--r-- 1 grid oinstall 52630342 May 24 09:28 ocssd.l01
-rw-r--r-- 1 grid oinstall 52631859 May 23 19:06 ocssd.l02
-rw-r--r-- 1 grid oinstall 52631497 May 23 04:28 ocssd.l03
-rw-r--r-- 1 grid oinstall 52634557 May 22 13:38 ocssd.l04
```

8. Compare the differences between the `day_ocr.xml` file and the `ocr_current_dump.xml` file to determine all changes made to the OCR in the last 24 hours. Log out as `root` when finished.

```
[root@host01 ~]# diff /tmp/day_ocr.xml
/tmp/ocr_current_dump.xml|more
3,5c3,4
< <TIMESTAMP>05/24/2013 19:36:42</TIMESTAMP>
< <DEVICE>/home/oracle/labs/day.ocr</DEVICE>
< <COMMAND>/u01/app/12.1.0/grid/bin/ocrdump.bin -xml -backupfile
/home/oracle/labs/day.ocr /home/oracle/labs/day_ocr.xml </COMMAND>
---
> <TIMESTAMP>05/24/2013 19:18:13</TIMESTAMP>
> <COMMAND>/u01/app/12.1.0/grid/bin/ocrdump.bin -xml
/home/oracle/labs/ocr_current_dump.xml </COMMAND>
230a230,241
> <KEY>
> <NAME>SYSTEM.WALLET.CVUDB_orcl</NAME>
> <VALUE_TYPE>BYTESTREAM (16)</VALUE_TYPE>
>
<VALUE><![CDATA[a1f84e370000000060000002106e18bed0b6258cce1164a639810
e99e85ca035ebdf8c0c37e5737e6bb3c01cb0530820b8402010330820b4a060
92a864886f70d010701a0820b3b04820b3730820b3330820b2f06092a864886f70d0
10706a0820b2030820b1c02010030820b1506092a864886f70d010701301c060a
2a864886f70d010c0103300e04085adca83480de844d020207d080820ae851fd402d
e6cf213c78dd915cb70a8791427b18e550378f0f7e0e868a2519981faa16f6388
dac16810b5a61dbfe1168a9faaa3a5296277becc121e0bfe8e7ed425e359244d179e
aefb486ddad540e5ef517dea6712214addcfad75d753f8da3d2379c6c4c697d82
bb5df613a6f993d45e58a3178d2e605e614c75c2615574bb3a3856c7fc461a5c8386
3f410e0f8f9ce30732e76ac96d1a6703860fc4528848879ac7dcd1650c23fc579
0a043d921f49cdf379a9eb829f327f0152c6b3519610615527002bcb7665b0cc364c
4e6209189d39ac1e1ef10352d9b11795449a47db6b277fcaa3256481efe2c5f7f
3e4f5ff2763b338b2206a1a3394467028bf610431b9c0b40b3273305ac3fedbd828c
0eec3f5525b2c3c2d58500286c5d2cc8b2b65d45d68779b6cb4da233247cee10c
a3ad3abfa2a83550f20d73549ac4e5058ebb20dcf68cd8a61773bfaf3476f1bb58fa
2071e4d8edc927fe12f95656440216e4b3195d08470e5ae4de7025a21fef6e1c3
f9bb603e8cb0812b49492ef125c2cf5f2f82934fad156ac709661113c2265f040912
b100bef2733b4d8469b98a5d0d5b03b142c03df16c5a75e622937c7e563f46d74
9138d72d3e6b84ac010d419a219462ebdb7d9d56baa47bb32f10795ba242278d1601
a516916fcdb1cf159ad9dde926bdd26f4406fd12b420985bbff1dac3c882e0593
```

```
5e18ff6d5b0e1f6769dceeef4529181286171bf6eb7928d9f3d73fc725468d7b2473
33314f74051a025a9ab5262467357c68a3898d61c34c1f823f1cbacb83f2279f6
e53b4d26481c24f9e10c6bf6981c07605d5e4aa5f47161533d014c32d02f1c4e811f
e585b5d70bba48cc03fc6533b8b6b801533cab509378db155b006bdc293ffe3fa
2c9c6f0e36333338f6bc5d5a70acae6439277418872ef2fc47daf39b68f6c136c407
ff07c4e3cd44f5faca1a9d175533082777ff749c001027b4af076170557afa8e4
109582fd63eb3eb3ba360c7b641b41b5a1f5e999e2dd8d2956e074f8ab88fff91b0b
cb10f7ee763c228e3cde3f75ecba677ed04c372315fb89f1413bf0005f6b3b07a
594dd48025f029c4859710ff88e9afeb8891e65e82bbd444427cac44d1cddc5ef095
d88b1d01ee8f14a6a95e7cd098044ec7cb8c8c98ebb1fe3710d4eb3e951fcc063
b7fdeb41bd30e13cdbb9d5185cc4741edc45faede19bb877a7cef5bf19af4e2cfe9f
6904d5977cb50ebb9c5b26f95437557a83cde534978370a168e90b167e44a27cf
ec41d8247f859f4c920b26a0aad307d309a3d095d6d40a13a836c02ea09438339668
d6c3c1865dda90133205f0167fb4f5e27f4b382ffe2ddbbc6f88d6b6b88635ef4
4aa7fb3bf385ce6f7e4688b352b67a5afe0d58bf6bdecfccb4b7496b4c4d659f7906
57a0e866ca658230062a7b626eb0983f934650a87693a141760313fb839c79473
a67bd2ab90ecaa2ba39e18cb4cc48b7481f4b095d7ab3c7c41e28b45121f73669393
6a0d738e34272bab4583dcb93dfc028aae9a96e43a848644e28fb774261cf60c7
2d7f3b3631aa6f204c5337a234bdee4ee454401867b22f42b2b3755eb1bd1aba93de
21e9440f0bb1f4144c02df864112a27c3cf6afe47fbd789dfbd8812faa5d4bdfe
f94cd66d2f9ab4c8bf83a4f3f0524274208d750da51ce4bbf8f2fe73ed18ad49b7f9
e4edd9cf1c16dd2b811fbfae174b17c5449777554eb6bd2098775a5877edfc89d
f19e1498929765164911 3baffb8217ba9995750fe223a31bf07ef1d6dcbee8adad59
682c2eee77dc987732d5bc1db7fa89a936c3b1f1d8cb14b35b896c3b1290f15ca
8bfadb3aca8aafed77b75e57deb8e4e381c90c4d738b8f16c4bf8987249e7fa9449f
d1b6019e7e6f5813e7dab3de5d7fb2e97895942f6ca6e425b2c49ef99b7bfdba6
dd4aa649eb533f45b2129eddd347261de54fbcecfff32ae6d0351510cb64577739a2
f9d3f2c77ee69e338b3d5e154da0c204066251ad8cd445aa2554c45d600827d0a
306277da7d2b553806c94231e8a77c0a3e230276ced632ee514bbce40278ad537d7e
2bd89dcd0f4b614a3cc588e64f80774682d128e3744560d66a452089c26a8fe63
941ea536a2f2a79be3b003ab9dd2dc37a06d2830ab9c8d0e6aecb51f744a78776396
b011185f811440de2e2ac9f7b90075c7670f4ca15d6c7f5e2a2c2208b54826d41
c736d071dee98d85d164b4f4610612d538aad23c5a2b1d4f64bbc15b887200213e29
7ccb58fd01b18ca8057352b8b3b02dd8ac0bb45abfb8b3588adbce0d15b68bf89
2578c39754aaff4604e65a7f2ce2feb52a6ffdd39afbabb89db043e27c029e2796b8
a6390717b3f9c27282666ba42549f4779e9f72403e68786a3e330a26711c722bd
9fa93c07fc7a0f7463bbe84266f8419927391fb3cd41cde8e6554eb4d910958c3061
e05e695d58584a83104417451a4bf58ea541561579f9fb9f4f09d8d4baee4a535
...
[root@host01 ~]# exit
logout
[grid@host01 cssd]$ cd
[grid@host01 ~]$
```

Practice 10-3: Working with CLUVFY

Overview

In this practice, you will work with CLUVFY to verify the state of various cluster components.

1. Determine the location of the cluvfy utility and its configuration file.

```
[grid@host01 ~]$ which cluvfy
/u01/app/12.1.0/grid/bin/cluvfy

[grid@host01 ~]$ cd $ORACLE_HOME/cv/admin

[grid@host01 admin]$ cat cvu_config
# Configuration file for Cluster Verification Utility(CVU)
# Version: 011405
#
# NOTE:
# 1._ Any line without a '=' will be ignored
# 2._ Since the fallback option will look into the environment
variables,
#     please have a component prefix(CV_) for each property to
define a
#     namespace.
#


#Nodes for the cluster. If CRS home is not installed, this list will
be
#picked up when -n all is mentioned in the commandline argument.
#CV_NODE_ALL=

#if enabled, cvuqdisk rpm is required on all nodes
CV_RAW_CHECK_ENABLED=TRUE

# Fallback to this distribution id
CV_ASSUME_DISTID=OEL4

#Complete file system path of sudo binary file, default is
/usr/local/bin/sudo
CV_SUDO_BINARY_LOCATION=/usr/local/bin/sudo

#Complete file system path of pbrun binary file, default is
/usr/local/bin/pbrun
CV_PBRUN_BINARY_LOCATION=/usr/local/bin/pbrun

# Whether X-Windows check should be performed for user equivalence
with SSH
#CV_XCHK_FOR_SSH_ENABLED=TRUE

# To override SSH location
#ORACLE_SRVM_REMOTESHELL=/usr/bin/ssh
```

```
# To override SCP location
#ORACLE_SRVM_REMOTECOPY=/usr/bin/scp

# To override version used by command line parser
CV_ASSUME_CL_VERSION=12.1

# Location of the browser to be used to display HTML report
#CV_DEFAULT_BROWSER_LOCATION=/usr/bin/mozilla
[grid@host01 admin]$
```

2. Display the stage options and stage names that can be used with the `cluvfy` utility.

```
[grid@host01 admin]$ cluvfy stage -list

USAGE:
cluvfy stage {-pre|-post} <stage-name> <stage-specific options>  [-
verbose]

Valid Stages are:
      -pre cfs        : pre-check for CFS setup
      -pre crsinst    : pre-check for CRS installation
      -pre acfscfg    : pre-check for ACFS Configuration.
      -pre dbinst     : pre-check for database installation
      -pre dbcfg      : pre-check for database configuration
      -pre hacfg      : pre-check for HA configuration
      -pre nodeadd    : pre-check for node addition.
      -post hwos      : post-check for hardware and operating system
      -post cfs       : post-check for CFS setup
      -post crsinst   : post-check for CRS installation
      -post acfscfg   : post-check for ACFS Configuration.
      -post hacfg     : post-check for HA configuration
      -post nodeadd   : post-check for node addition.
      -post nodedel   : post-check for node deletion.

[grid@host01 admin]$
```

3. Perform a postcheck for the ACFS configuration on all nodes.

```
[grid@host01 admin]$ cluvfy stage -post acfscfg -n all

Performing post-checks for ACFS Configuration

Checking node reachability...
Node reachability check passed from node "host01"


Checking user equivalence...
User equivalence check passed for user "grid"
Task ASM Integrity check started...

```

```
Checking if connectivity exists across cluster nodes on the ASM
network

Checking node connectivity...

Checking hosts config file...

Verification of the hosts config file successful

Check: Node connectivity using interfaces on subnet "192.168.2.0"
Node connectivity passed for subnet "192.168.2.0" with node(s)
host01,host02,host03
TCP connectivity check passed for subnet "192.168.2.0"


Check: Node connectivity using interfaces on subnet "192.168.1.0"
Node connectivity passed for subnet "192.168.1.0" with node(s)
host02,host03,host01
TCP connectivity check passed for subnet "192.168.1.0"

Checking subnet mask consistency...
Subnet mask consistency check passed for subnet "192.168.1.0".
Subnet mask consistency check passed for subnet "192.168.2.0".
Subnet mask consistency check passed.

Node connectivity check passed

Network connectivity check across cluster nodes on the ASM network
passed

Starting check to see if ASM is running on all cluster nodes...

ASM Running check passed. ASM is running on all specified nodes

Starting Disk Groups check to see if at least one Disk Group
configured...
Disk Group Check passed. At least one Disk Group configured

Task ASM Integrity check passed...

Task ACFS Integrity check started...

Task ACFS Integrity check passed

UDev attributes check for ACFS started...
UDev attributes check passed for ACFS


Post-check for ACFS Configuration was successful.
[grid@host01 admin]$ $
```

4. Display a list of the component names that can be checked with the `cluvfy` utility.

```
 [grid@host01 admin]$ cluvfy comp -list

USAGE:
cluvfy comp  <component-name> <component-specific options>  [-
verbose]

Valid Components are:
      nodereach       : checks reachability between nodes
      nodecon         : checks node connectivity
      cfs             : checks CFS integrity
      ssa             : checks shared storage accessibility
      space           : checks space availability
      sys             : checks minimum system requirements
      clu             : checks cluster integrity
      clumgr          : checks cluster manager integrity
      ocr             : checks OCR integrity
      olr             : checks OLR integrity
      ha              : checks HA integrity
      freespace       : checks free space in CRS Home
      crs             : checks CRS integrity
      nodeapp         : checks node applications existence
      admprv          : checks administrative privileges
      peer            : compares properties with peers
      software        : checks software distribution
      acfs            : checks ACFS integrity
      asm             : checks ASM integrity
      gpnp            : checks GPnP integrity
      gns             : checks GNS integrity
      scan            : checks SCAN configuration
      ohasd           : checks OHASD integrity
      clocksync       : checks Clock Synchronization
      vdisk           : checks Voting Disk configuration and UDEV
settings
      healthcheck     : checks mandatory requirements and/or best
practice recommendations
      dhcp            : checks DHCP configuration
      dns             : checks DNS configuration
      baseline        : collect and compare baselines

 [grid@host01 admin]$
```

5. Display the syntax usage help for the space component check of the cluvfy utility.

```
 [grid@host01 admin]$ cluvfy comp space -help

USAGE:
cluvfy comp space [-n <node_list>] -l <storage_location> -z
<disk_space>{B|K|M|G} [-verbose]
```

```
<node_list> is the comma-separated list of non-domain qualified node
names on which the test should be conducted. If "all" is specified,
then all the nodes in the cluster will be used for verification.
<storage_location> is the storage path.
<disk_space> is the required disk space, in units of
bytes(B),kilobytes(K),megabytes(M) or gigabytes(G).

DESCRIPTION:
Checks for free disk space at the location provided by '-l' option
on all the nodes in the nodelist. If no '-n' option is given, local
node is used for this check.

 [grid@host01 admin]$
```

6. Verify that on each node of the cluster the `/tmp` directory has at least 200 MB of free space in it using the `cluvfy` utility. Use verbose output.

```
 [grid@host01 admin]$ cluvfy comp space -n host01,host02,host03 -l
/tmp -z 200M -verbose

Verifying space availability

Checking space availability...

Check: Space available on "/tmp"

  Node Name   Available                Required                 Status
  ----------  ---------------------  -----------------------  ------
----
  host03      7.2478GB (7599920.0KB)    200MB (204800.0KB)     passed
   host02    7.2418GB (7593584.0KB)    200MB (204800.0KB)      passed
   host01    7.7335GB (8109208.0KB)    200MB (204800.0KB)      passed
Result: Space availability check passed for "/tmp"

Verification of space availability was successful.
 [grid@host01 admin]$
```

Practices for Lesson 11: Making Applications Highly Available

Chapter 11

Practices for Lesson 11: Overview

Practices Overview

In this practice, you will:

- Configure highly available application resources on Flex Cluster leaf nodes.
- Use Oracle Clusterware to protect the Apache application.

Practice 11-1: Configuring Highly Available Application Resources on Flex Cluster Leaf Nodes

Overview

In this practice you will create a series of highly available application resources running on one of the Flex Cluster Leaf Nodes.

1. Establish a terminal session connected to `host01` using the `grid` user. Configure the environment with the `oraenv` script.

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


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

[grid@host01 ~]$
```

2. Examine the cluster to identify the role for each node in the cluster.

```
[grid@host01 ~]$ crsctl get node role status -all
Node 'host01' active role is 'hub'
Node 'host02' active role is 'hub'
Node 'host03' active role is 'hub'
Node 'host04' active role is 'leaf'
Node 'host05' active role is 'leaf'
[grid@host01 ~]$
```

3. Examine the cluster to identify the currently defined server pools and server allocations. Notice that currently the Hub Nodes are allocated to the `orcldb` server pool which contains the RAC database. Also, both Leaf Nodes are currently in the Free pool.

```
[grid@host01 ~]$ crsctl status serverpool
NAME=Free
ACTIVE_SERVERS=host04 host05

NAME=Generic
ACTIVE_SERVERS=

NAME=bigpool
ACTIVE_SERVERS=

NAME=ora.orcldb
ACTIVE_SERVERS=host01 host02 host03

[grid@host01 ~]$
```

4. Examine the cluster to identify the currently defined server categories. Note the existence of the built-in category `ora.leaf.category`. All the Leaf Nodes in the cluster are implicitly associated with this category.

```
[grid@host01 ~]$ crsctl status category
NAME=big
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r--
ACTIVE_CSS_ROLE=hub
EXPRESSION=(MEMORY_SIZE > 8000)

NAME=ora.hub.category
ACL=owner:root:rwx,pgrp:root:r-x,other::r--
ACTIVE_CSS_ROLE=hub
EXPRESSION=

NAME=ora.leaf.category
ACL=owner:root:rwx,pgrp:root:r-x,other::r--
ACTIVE_CSS_ROLE=leaf
EXPRESSION=

[grid@host01 ~]$
```

5. Create a new server pool to support a series of highly available application resources on one of the Flex Cluster Leaf Nodes.

```
[grid@host01 ~]$ srvctl add srvpool -serverpool my_app_pool -min 1 -
max 1 -category "ora.leaf.category"
[grid@host01 ~]$
```

6. Reexamine the server pools. Notice that one of the leaf nodes has been allocated to the newly created server pool (`my_app_pool`).

```
[grid@host01 ~]$ crsctl status serverpool
NAME=Free
ACTIVE_SERVERS=host05

NAME=Generic
ACTIVE_SERVERS=

NAME=bigpool
ACTIVE_SERVERS=

NAME=ora.my_app_pool
ACTIVE_SERVERS=host04

NAME=ora.orcldb
ACTIVE_SERVERS=host01 host02 host03

[grid@host01 ~]$
```

7. Navigate to the directory that contains the scripts for this practice.

```
[grid@host01 ~]$ cd /stage/GRID/labs/less_11
[grid@host01 less_11]$
```

8. Examine the provided action script (`action.scr`). You will use this script to create a series of highly available application resources on a Flex Cluster Leaf Node. The application resources will all perform the following set of simple tasks.
   - When the application resource is started it creates an empty file at `/tmp/<resource_name>` on the node running the application resource.
   - When the application resource is stopped or cleaned the file at `/tmp/<resource_name>` is deleted.
   - When the application resource is checked a test is performed to check the existence of the file at `/tmp/<resource_name>`.
   - In addition, regardless of the action performed, log entries are written to the CRSD agent log file, which can be found at `/u01/app/12.1.0/grid/log/<hostname>/agent/crsd/scriptagent_grid/scriptagent_grid.log`.

```
[grid@host01 less_11]$ cat action.scr
#!/bin/sh
TOUCH=/bin/touch
RM=/bin/rm
PATH_NAME=/tmp/$_CRS_NAME

#
# These messages go into the CRSD agent log file.
echo " *******   `date` ********** "
echo "Action script '$_CRS_ACTION_SCRIPT' for resource[$_CRS_NAME]
called for action $1"
#

case "$1" in
  'start')
     echo "START entry point has been called.."
     echo "Creating the file: $PATH_NAME"
     $TOUCH $PATH_NAME
     exit 0
     ;;

  'stop')
     echo "STOP entry point has been called.."
     echo "Deleting the file: $PATH_NAME"
     $RM $PATH_NAME
     exit 0
     ;;

  'check')
    echo "CHECK entry point has been called.."
    if [ -e $PATH_NAME ]; then
        echo "Check -- SUCCESS"
        exit 0
    else
```

```
        echo "Check -- FAILED"
        exit 1
    fi
    ;;

  'clean')
     echo "CLEAN entry point has been called.."
     echo "Deleting the file: $PATH_NAME"
     $RM -f $PATH_NAME
     exit 0
     ;;

esac

[grid@host01 less_11]$
```

9. Examine the file `create_res.sh`. This file contains the commands which you will next use to create three application resources named `my_dep_res1`, `my_dep_res2`, and `my_resource`. Notice that all three resources are associated with the `my_app_pool` server pool. Notice also that `my_resource` has a mandatory (hard) dependency on `my_dep_res1`, and an optional (soft) dependency on `my_dep_res2`. Finally notice that `my_resource` also depends on the RAC database (`ora.orcl.db`). This illustrates how you can unify the management of databases and applications within a Flex Cluster.

```
[grid@host01 less_11]$ cat create_res.sh
crsctl add resource my_dep_res1 -type cluster_resource -attr
"ACTION_SCRIPT=/stage/GRID/labs/less_11/action.scr,PLACEMENT=restric
ted,SERVER_POOLS=ora.my_app_pool"

crsctl add resource my_dep_res2 -type cluster_resource -attr
"ACTION_SCRIPT=/stage/GRID/labs/less_11/action.scr,PLACEMENT=restric
ted,SERVER_POOLS=ora.my_app_pool"

crsctl add resource my_resource -type cluster_resource -attr
"ACTION_SCRIPT=/stage/GRID/labs/less_11/action.scr,PLACEMENT=restric
ted,SERVER_POOLS=ora.my_app_pool,START_DEPENDENCIES='hard(my_dep_res
1, global:uniform:ora.orcl.db) pullup:always(my_dep_res1,
global:ora.orcl.db)
weak(my_dep_res2)',STOP_DEPENDENCIES='hard(my_dep_res1,
global:ora.orcl.db)'"
[grid@host01 less_11]$
```

10. Execute `create_res.sh` to create the application resources.

```
[grid@host01 less_11]$ ./create_res.sh
[grid@host01 less_11]$
```

11. Examine the newly created cluster resources. Note that currently the resources exist but have not been started.

```
[grid@host01 less_11]$ crsctl status resource -t -w "NAME co my"
--------------------------------------------------------------------
```

```
Name           Target  State        Server             State details
--------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------
my_dep_res1
      1        OFFLINE OFFLINE                               STABLE
my_dep_res2
      1        OFFLINE OFFLINE                               STABLE
my_resource
      1        OFFLINE OFFLINE                               STABLE
--------------------------------------------------------------------
[grid@host01 less_11]$
```

12. Establish another terminal session connected to `host01` as the `oracle` user. Configure the oracle environment setting the `ORACLE_SID` variable to `orcl`.

```
[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 ~]$
```

13. Stop the `orcl` database so that you can exercise the dependency between `my_resource` and your RAC database. Confirm the database has been stopped on all nodes.

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

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

[oracle@host01 ~]$
```

14. Back in your `grid` terminal session, start the `my_resource` application resource. Notice that `my_dep_res1` and `my_dep_res2` are started automatically to fulfill the dependency definitions, and that all three resources are started on the server associated with the `my_app_pool` server pool. Note also that the `orcl` database is also started as defined in the dependency definitions for the `my_resource` application resource. This illustrates how a Flex Cluster can be used to manage a RAC database and associated application resources.

```
[grid@host01 less_11]$ crsctl start resource my_resource
CRS-2672: Attempting to start 'my_dep_res1' on 'host04'
CRS-2672: Attempting to start 'my_dep_res2' on 'host04'
CRS-2672: Attempting to start 'ora.orcl.db' on 'host03'
CRS-2672: Attempting to start 'ora.orcl.db' on 'host01'
CRS-2672: Attempting to start 'ora.orcl.db' on 'host02'
CRS-2676: Start of 'my_dep_res1' on 'host04' succeeded
CRS-2676: Start of 'my_dep_res2' on 'host04' succeeded
```

```
CRS-2676: Start of 'ora.orcl.db' on 'host03' succeeded
CRS-2676: Start of 'ora.orcl.db' on 'host01' succeeded
CRS-2676: Start of 'ora.orcl.db' on 'host02' succeeded
CRS-2672: Attempting to start 'my_resource' on 'host04'
CRS-2676: Start of 'my_resource' on 'host04' succeeded
[grid@host01 less_11]$
```

15. Re-examine the application resources to confirm their status.

```
[grid@host01 less_11]$ crsctl status resource -t -w "NAME co my"
--------------------------------------------------------------------
Name           Target  State        Server             State details
--------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------
my_dep_res1
      1        ONLINE  ONLINE       host04                   STABLE
my_dep_res2
      1        ONLINE  ONLINE       host04                   STABLE
my_resource
      1        ONLINE  ONLINE       host04                   STABLE
--------------------------------------------------------------------
[grid@host01 less_11]$
```

16. Check the files under `/tmp` on the node running the application resources. You should expect to see a series of empty files, each bearing the name of one of the application resources.

```
[grid@host01 less_11]$ ssh host04 ls -la /tmp/my*
-rw-r--r-- 1 grid oinstall 0 May 30 19:04 /tmp/my_dep_res1
-rw-r--r-- 1 grid oinstall 0 May 30 19:04 /tmp/my_dep_res2
-rw-r--r-- 1 grid oinstall 0 May 30 19:06 /tmp/my_resource
[grid@host01 less_11]$
```

17. From the `oracle` terminal session, confirm also that your RAC database (`orcl`) started as part of starting the `my_resource` application resource.

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

18. Switch user (`su`) to `root` and shutdown Oracle Clusterware on the node hosting the application resources. Don't forget to set the Grid environment for `root`.

```
[grid@host01 less_11]$ su -
Password:
[root@host01 ~]# . oraenv
ORACLE_SID = [root] ? +ASM1
The Oracle base has been set to /u01/app/grid
```

```
[root@host01 ~]# crsctl stop cluster -n host04
CRS-2673: Attempting to stop 'ora.crsd' on 'host04'
CRS-2790: Starting shutdown of Cluster Ready Services-managed
resources on 'host04'
CRS-2673: Attempting to stop 'my_resource' on 'host04'
CRS-2673: Attempting to stop 'my_dep_res2' on 'host04'
CRS-2677: Stop of 'my_resource' on 'host04' succeeded
CRS-2673: Attempting to stop 'my_dep_res1' on 'host04'
CRS-2677: Stop of 'my_dep_res2' on 'host04' succeeded
CRS-2677: Stop of 'my_dep_res1' on 'host04' succeeded
CRS-2792: Shutdown of Cluster Ready Services-managed resources on
'host04' has completed
CRS-2677: Stop of 'ora.crsd' on 'host04' succeeded
CRS-2673: Attempting to stop 'ora.cluster_interconnect.haip' on
'host04'
CRS-2673: Attempting to stop 'ora.ctssd' on 'host04'
CRS-2673: Attempting to stop 'ora.evmd' on 'host04'
CRS-2673: Attempting to stop 'ora.storage' on 'host04'
CRS-2677: Stop of 'ora.storage' on 'host04' succeeded
CRS-2677: Stop of 'ora.cluster_interconnect.haip' on 'host04'
succeeded
CRS-2677: Stop of 'ora.ctssd' on 'host04' succeeded
CRS-2677: Stop of 'ora.evmd' on 'host04' succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host04'
CRS-2677: Stop of 'ora.cssd' on 'host04' succeeded
[root@host01 ~]#
```

19. Wait a few moments, then examine the status of the server pools. You should see that the previously unallocated Leaf Node has been moved to the `my_app_pool` server pool. Exit the `root` account when finished

```
[root@host01 ~]# crsctl status serverpool
NAME=Free
ACTIVE_SERVERS=

NAME=Generic
ACTIVE_SERVERS=

NAME=bigpool
ACTIVE_SERVERS=

NAME=ora.my_app_pool
ACTIVE_SERVERS=host05

NAME=ora.orcldb
ACTIVE_SERVERS=host01 host02 host03

[root@host01 ~]# exit
logout
[grid@host01 less_11]$
```

20. Re-examine the status of the application resources. Notice that they have been failed-over to the surviving Leaf Node.

```
[grid@host01 less_11]$ crsctl status resource -t -w "NAME co my"
--------------------------------------------------------------------
Name           Target  State        Server             State details
--------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------
my_dep_res1
      1        ONLINE  ONLINE       host05                   STABLE
my_dep_res2
      1        ONLINE  ONLINE       host05                   STABLE
my_resource
      1        ONLINE  ONLINE       host05                   STABLE
--------------------------------------------------------------------
[grid@host01 less_11]$
```

21. Re-examine the contents of the `/tmp` directory on both Leaf Nodes. Notice that the files you saw in step 16 no longer exist and that there is newer set of files on the other Leaf Node.

```
[grid@host01 less_11]$ ssh host04 ls -la /tmp/my*
ls: cannot access /tmp/my*: No such file or directory

[grid@host01 less_11]$ ssh host05 ls -la /tmp/my*
-rw-r--r-- 1 grid oinstall 0 May 30 19:18 /tmp/my_dep_res1
-rw-r--r-- 1 grid oinstall 0 May 30 19:18 /tmp/my_dep_res2
-rw-r--r-- 1 grid oinstall 0 May 30 19:18 /tmp/my_resource
[grid@host01 less_11]$
```

22. Restart Oracle Clusterware as the `root` user on the inactive Leaf Node.

```
[grid@host01 less_11]$ su -
Password:

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

[root@host01 ~]# crsctl start cluster -n host04
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'host04'
CRS-2672: Attempting to start 'ora.evmd' on 'host04'
CRS-2676: Start of 'ora.cssdmonitor' on 'host04' succeeded
CRS-2672: Attempting to start 'ora.cssd' on 'host04'
CRS-2672: Attempting to start 'ora.diskmon' on 'host04'
CRS-2676: Start of 'ora.diskmon' on 'host04' succeeded
CRS-2676: Start of 'ora.evmd' on 'host04' succeeded
CRS-2676: Start of 'ora.cssd' on 'host04' succeeded
CRS-2672: Attempting to start 'ora.cluster_interconnect.haip' on
'host04'
CRS-2672: Attempting to start 'ora.storage' on 'host04'
CRS-2672: Attempting to start 'ora.ctssd' on 'host04'
```

```
CRS-2676: Start of 'ora.storage' on 'host04' succeeded
CRS-2676: Start of 'ora.cluster_interconnect.haip' on 'host04'
succeeded
CRS-2676: Start of 'ora.ctssd' on 'host04' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'host04'
CRS-2676: Start of 'ora.crsd' on 'host04' succeeded
[root@host01 ~]#
```

23. Check server pool status. Check and see what pool the server you just started the Clusterware stack on currently belongs to.

```
[root@host01 ~]# crsctl status serverpool
NAME=Free
ACTIVE_SERVERS=host04

NAME=Generic
ACTIVE_SERVERS=

NAME=bigpool
ACTIVE_SERVERS=

NAME=ora.my_app_pool
ACTIVE_SERVERS=host05

NAME=ora.orcldb
ACTIVE_SERVERS=host01 host02 host03

[root@host01 ~]#
```

Congratulations! You have successfully configured highly available application resources on Flex Cluster Leaf Nodes

24. To finish, please exit the root account. Stop and delete the `my_resource`, `my_dep_res1`, and `my_dep_res2` resources. Delete the `ora.my_app_pool` server pool. Exit the terminal session.

```
[root@host01 ~]# exit
logout
[grid@host01 less_11]$ crsctl stop resource my_resource
CRS-2673: Attempting to stop 'my_resource' on 'host05'
CRS-2677: Stop of 'my_resource' on 'host05' succeeded

[grid@host01 less_11]$ crsctl stop resource my_dep_res1
CRS-2673: Attempting to stop 'my_dep_res1' on 'host05'
CRS-2677: Stop of 'my_dep_res1' on 'host05' succeeded

[grid@host01 less_11]$ crsctl stop resource my_dep_res2
CRS-2673: Attempting to stop 'my_dep_res2' on 'host05'
CRS-2677: Stop of 'my_dep_res2' on 'host05' succeeded

[grid@host01 less_11]$ crsctl delete resource my_resource

[grid@host01 less_11]$ crsctl delete resource my_dep_res1
```

```
[grid@host01 less_11]$ crsctl delete resource my_dep_res2

[grid@host01 less_11]$ crsctl delete serverpool ora.my_app_pool

[grid@host01 less_11]$ exit
logout
Connection to host01 closed.
$
```

Practice 11-2: Protecting the Apache Application

Overview

In this practice, you use Oracle Clusterware to protect the Apache application. To do this, you create an application VIP for Apache (HTTPD), an action script, and a resource.

1. As the `root` user, verify that the Apache RPMs, `httpd`, and `httpd-tools`, are installed on `host04` and `host05`.

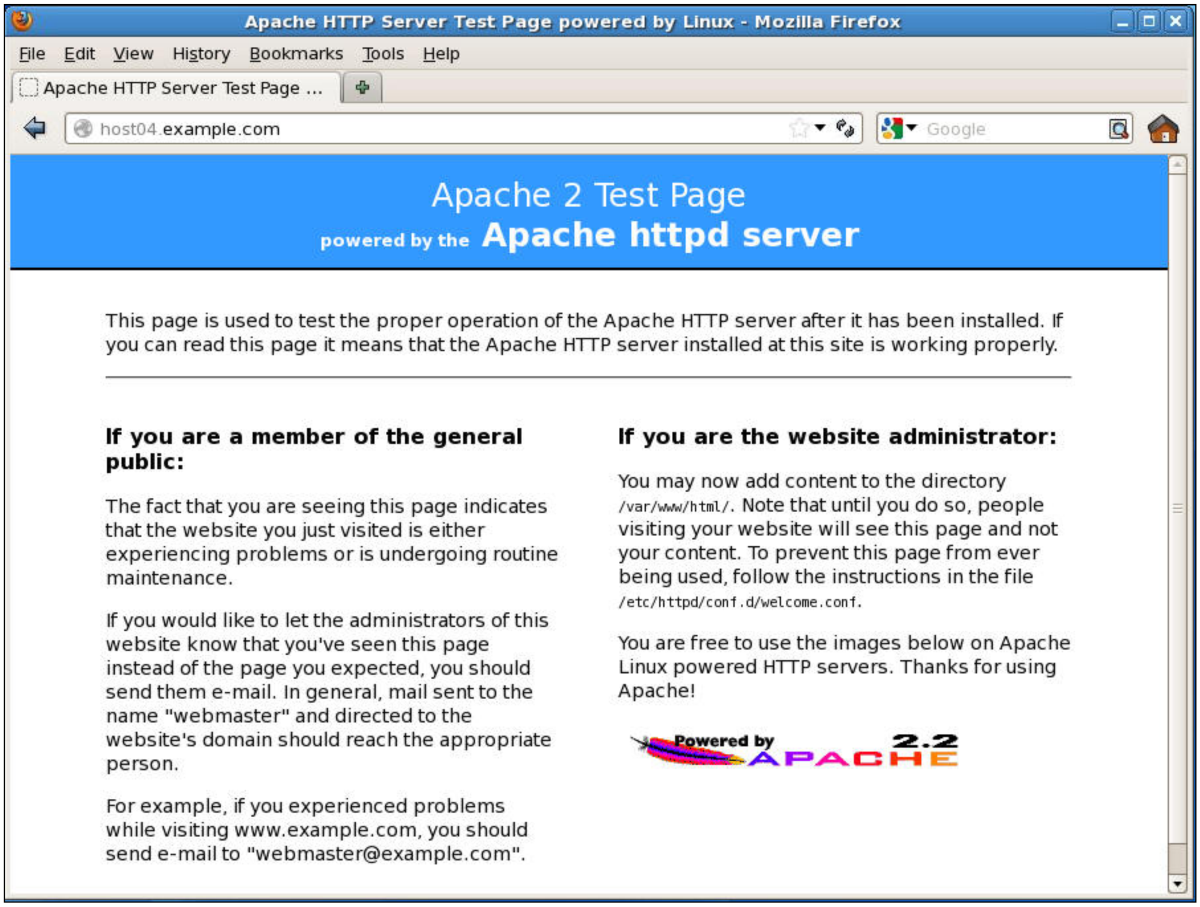
```
$ ssh root@host04
Last login: Thu May 30 12:02:16 2013 from 192.0.2.1
[root@host04 ~]# rpm -qa|grep httpd
httpd-2.2.15-15.0.1.el6_2.1.x86_64
httpd-tools-2.2.15-15.0.1.el6_2.1.x86_64

[root@host04 ~]# ssh host05 rpm -qa|grep httpd
httpd-2.2.15-15.0.1.el6_2.1.x86_64
httpd-tools-2.2.15-15.0.1.el6_2.1.x86_64
[root@host04 ~]#
```

2. As the `root` user, start the Apache application on `host04` with the `apachectl start` command:

```
[root@host04 ~]# apachectl start
[root@host04 ~]#
```

Open a browser on your desktop and access the Apache test page on `host04`. The HTTP address would look something like this:

http://host04.example.com



Apache HTTP Server Test Page powered by Linux - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Apache HTTP Server Test Page ...
host04.example.com
Google

Apache 2 Test Page

powered by the **Apache httpd server**

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the images below on Apache Linux powered HTTP servers. Thanks for using Apache!

After you have determined that Apache is working properly, stop Apache using the `apachectl stop` command.

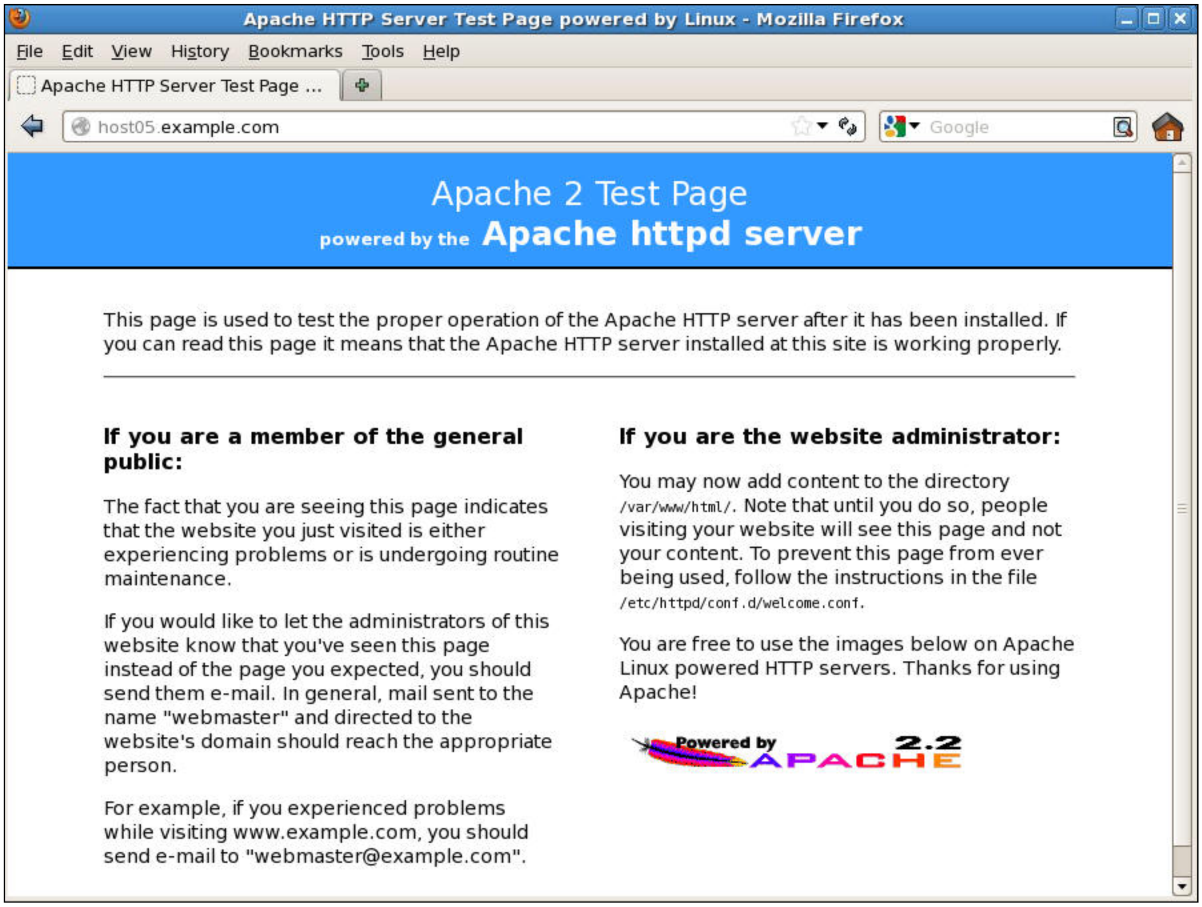
```
[root@host04 ~]# apachectl stop
[root@host04 ~]#
```

3. As the `root` user, start the Apache application on `host04` with the `apachectl start` command.

```
[root@host04 ~]# ssh host05 apachectl start
[root@host04 ~]#
```

Open a browser on your desktop and access the Apache test page on `host04`. The HTTP address would look something like this:

http://host05.example.com


Apache HTTP Server Test Page powered by Linux - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Apache HTTP Server Test Page ...
host05.example.com
Google

Apache 2 Test Page
powered by the **Apache httpd server**

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the images below on Apache Linux powered HTTP servers. Thanks for using Apache!

After you have determined that Apache is working properly, stop Apache using the `apachectl stop` command.

```
[root@host04 ~]# ssh host05 apachectl stop
[root@host04 ~]#
```

4. Create an action script to control the application. This script must be accessible by all nodes on which the application resource can be located. As the `root` user, create a script on `host04` called `apache.scr` in `/usr/local/bin` that will start, stop, check status, and clean up if the application does not exit cleanly. **Make sure that the host specified in the `WEBPAGECHECK` variable is `host04`.** Use the `/stage/GRID/labs/less_11/apache.scr.tpl` file as a template for creating the script. Make the script executable and test the script.

```
[root@host04 ~]# cp /stage/GRID/labs/less_11/apache.tpl
/usr/local/bin/apache.scr
[root@host04 ~]# vi /usr/local/bin/apache.scr
#!/bin/bash

HTTPDCONFLOCATION=/etc/httpd/conf/httpd.conf
WEBPAGECHECK=http://host04.example.com:80/icons/apache_pb.gif
case $1 in
'start')
/usr/sbin/apachectl -k start -f $HTTPDCONFLOCATION
RET=$?
;;
'stop')
/usr/sbin/apachectl -k stop
RET=$?
;;
'clean')
/usr/sbin/apachectl -k stop
RET=$?
;;
'check')
/usr/bin/wget -q --delete-after $WEBPAGECHECK
RET=$?
;;
*)
RET=0
;;
esac
# 0: success; 1 : error
if [ $RET -eq 0 ]; then
exit 0
else
exit 1
fi
```

Save the file

```
[root@host04 ~]# chmod 755 /usr/local/bin/apache.scr

[root@host04 ~]# apache.scr start
```

Verify web page

```
[root@host04 ~]#  apache.scr stop
```

Refresh http://host04.example.com, Web page should no longer display

5. Let’s repeat the actions in the previous step now on `host05`. As the `root` user, log into `host05` and create a script called `apache.scr` in `/usr/local/bin` that will start, stop, check status, and clean up if the application does not exit cleanly. **Make sure that the host specified in the `WEBPAGECHECK` variable is `host05`**. Use the `/stage/GRID/labs/less_11/apache.scr.tpl` file as a template for creating the script. Make the script executable and test the script. When finished, exit `host05`, returning to `host04`.

```
[root@host04 ~]# ssh host05

[root@host05 ~]# cp /stage/GRID/labs/less_11/apache.tpl
/usr/local/bin/apache.scr

[root@host05 ~]# vi /usr/local/bin/apache.scr
#!/bin/bash

HTTPDCONFLOCATION=/etc/httpd/conf/httpd.conf
WEBPAGECHECK=http://host05.example.com:80/icons/apache_pb.gif
case $1 in
'start')
/usr/sbin/apachectl -k start -f $HTTPDCONFLOCATION
RET=$?
;;
'stop')
/usr/sbin/apachectl -k stop
RET=$?
;;
'clean')
/usr/sbin/apachectl -k stop
RET=$?
;;
'check')
/usr/bin/wget -q --delete-after $WEBPAGECHECK
RET=$?
;;
*)
RET=0
;;
esac
# 0: success; 1 : error
if [ $RET -eq 0 ]; then
exit 0
else
exit 1
fi
```

Save the file

```
[root@host05 ~]# chmod 755 /usr/local/bin/apache.scr

[root@host05 ~]# apache.scr start
```

Verify web page

```
[root@host05 ~]#  apache.scr stop
```

Refresh http://host05.example.com, Web page should no longer display

```
[root@host05 ~]# exit
[root@host04 ~]#
```

6. Next, validate the return code of a check failure using the new script. The Apache server should NOT be running on either node. Run `apache.scr check` and immediately test the return code by issuing an `echo $?` command. This must be run immediately after the `apache.scr check` command because the shell variable `$?` holds the exit code of the previous command run from the same shell. An unsuccessful check should return an exit code of `1`. You should do this on both nodes.

```
[root@host04 ~]# apache.scr check
[root@host04 ~]# echo $?
1

[root@host04 ~]# ssh host05

[root@host05 ~]# apache.scr check
[root@host05 ~]# echo $?
1

[root@host05 ~]# exit
logout
Connection to host05 closed.
[root@host04 ~]#
```

7. As the `grid` user, create a server pool for the resource called `myApache_sp`. This pool contains `host04` and `host05` and is a child of the Free pool.

```
[root@host04 ~]# su - grid

[grid@host04 ~]$ /u01/app/12.1.0/grid/bin/crsctl add serverpool
myApache_sp  -attr "PARENT_POOLS=Free, SERVER_NAMES=host04 host05"

[grid@host04 ~]$
```

8. Check the status of the new pool on your cluster. Exit the grid account when finished.

```
[grid@host04 ~]$ /u01/app/12.1.0/grid/bin/crsctl status serverpool
NAME=Free
ACTIVE_SERVERS=host04 host05

NAME=Generic
ACTIVE_SERVERS=

NAME=bigpool
ACTIVE_SERVERS=

NAME=myApache_sp
ACTIVE_SERVERS=host04 host05
```

```
NAME=ora.orcldb
ACTIVE_SERVERS=host01 host02 host03

[grid@host04 ~]$ exit
logout
[root@host04 ~]#
```

9. Add the Apache Resource, which will be called `myApache`, to the `myApache_sp` subpool. It must be performed as `root` as the resource requires `root` authority because of listening on the default privileged port 80. Set `CHECK_INTERVAL` to `30`, `RESTART_ATTEMPTS` to `2`, and `PLACEMENT` to `restricted`.

```
[root@host04 ~]# /u01/app/12.1.0/grid/bin/crsctl add resource
myApache -type cluster_resource -attr
"ACTION_SCRIPT=/usr/local/bin/apache.scr, PLACEMENT='restricted',
SERVER_POOLS=myApache_sp, CHECK_INTERVAL='30', RESTART_ATTEMPTS='2'"
[root@host04 ~]#
```

10. View the static attributes of the `myApache` resource.

```
[root@host04 ~]# /u01/app/12.1.0/grid/bin/crsctl status resource
myApache –f

NAME=myApache
TYPE=cluster_resource
STATE=OFFLINE
TARGET=OFFLINE
ACL=owner:root:rwx,pgrp:root:r-x,other::r--
ACTIONS=
ACTION_FAILURE_TEMPLATE=
ACTION_SCRIPT=/usr/local/bin/apache.scr
ACTION_TIMEOUT=60
ACTIVE_PLACEMENT=0
AGENT_FILENAME=%CRS_HOME%/bin/scriptagent
ALERT_TEMPLATE=
ALIAS_NAME=
AUTO_START=restore
CARDINALITY=1
CARDINALITY_ID=0
CHECK_INTERVAL=30
CHECK_TIMEOUT=0
CLEAN_TIMEOUT=60
CREATION_SEED=243
DEFAULT_TEMPLATE=
DEGREE=1
DELETE_TIMEOUT=60
DESCRIPTION=
ENABLED=1
FAILOVER_DELAY=0
FAILURE_INTERVAL=0
FAILURE_THRESHOLD=0
HOSTING_MEMBERS=
ID=myApache
INSTANCE_COUNT=1
INSTANCE_FAILOVER=1
INTERMEDIATE_TIMEOUT=0
LOAD=1
LOGGING_LEVEL=1
MODIFY_TIMEOUT=60
NOT_RESTARTING_TEMPLATE=
OFFLINE_CHECK_INTERVAL=0
PLACEMENT=restricted
PROFILE_CHANGE_TEMPLATE=
RELOCATE_BY_DEPENDENCY=1
RESTART_ATTEMPTS=2
SCRIPT_TIMEOUT=60
SERVER_CATEGORY=
SERVER_POOLS=myApache_sp
START_CONCURRENCY=0
START_DEPENDENCIES=
START_TIMEOUT=0
STATE_CHANGE_TEMPLATE=
STOP_CONCURRENCY=0
STOP_DEPENDENCIES=
STOP_TIMEOUT=0
UPTIME_THRESHOLD=1h
USER_WORKLOAD=no
USE_STICKINESS=0

[root@host04 ~]#
```

11. Start the new resource. Confirm that the resource is online on one of the leaf nodes. If you like, open a browser and point it to the node on which the `myApache` resource is placed.

```
[root@host04 ~]# /u01/app/12.1.0/grid/bin/crsctl start resource
myApache
CRS-2672: Attempting to start 'myApache' on 'host04'
CRS-2676: Start of 'myApache' on 'host04' succeeded

[root@host04 ~]# /u01/app/12.1.0/grid/bin/crsctl status resource
myApache

NAME=myApache
TYPE=cluster_resource
TARGET=ONLINE
STATE=ONLINE on host04

[root@host04 ~]#
```

Open the indicated host in a browser to verify the test page is displayed:

http://host04.example.com

12. Confirm that Apache is NOT running on the other leaf node. The easiest way to do this is to check for the running `/usr/sbin/httpd -k start -f`

/etc/httpd/conf/httpd.confd processes with the ps command. Check on the host where the resource is currently placed first. Then check the node where it should NOT be running.

```
[root@host04 ~]# ps -ef|grep -i "httpd -k"
root      16977     1  0 20:46 ?        00:00:00 /usr/sbin/httpd -k
start -f /etc/httpd/conf/httpd.conf
apache    16978 16977  0 20:46 ?        00:00:00 /usr/sbin/httpd -k
start -f /etc/httpd/conf/httpd.conf
apache    16979 16977  0 20:46 ?        00:00:00 /usr/sbin/httpd -k
start -f /etc/httpd/conf/httpd.conf
apache    16980 16977  0 20:46 ?        00:00:00 /usr/sbin/httpd -k
start -f /etc/httpd/conf/httpd.conf
apache    16981 16977  0 20:46 ?        00:00:00 /usr/sbin/httpd -k
start -f /etc/httpd/conf/httpd.conf
apache    16982 16977  0 20:46 ?        00:00:00 /usr/sbin/httpd -k
start -f /etc/httpd/conf/httpd.conf
apache    16986 16977  0 20:46 ?        00:00:00 /usr/sbin/httpd -k
start -f /etc/httpd/conf/httpd.conf
apache    16987 16977  0 20:46 ?        00:00:00 /usr/sbin/httpd -k
start -f /etc/httpd/conf/httpd.conf
apache    16988 16977  0 20:46 ?        00:00:00 /usr/sbin/httpd -k
start -f /etc/httpd/conf/httpd.conf
root      17081 16662  0 20:53 pts/0    00:00:00 grep -i httpd -k

[root@host04 ~]# ssh host05 ps -ef|grep -i "httpd -k"
[root@host04 ~]#
```

13. Simulate a node failure on the node hosting the myApache resource using the reboot command as root. Before issuing the reboot on the first node, open a terminal session on the second node and as the root user, execute the /stage/GRID/labs/less_11/monitor.sh script so that you can monitor the failover.

```
$ ssh root@host05
Last login: Thu May 30 21:08:47 2013 from 192.0.2.1
[root@host05 ~]#
```

Switch terminal windows to the node hosting the myApache resource

```
[root@host04 ~]# reboot

Broadcast message from root@host04
        (/dev/pts/0) at 21:11 ...

The system is going down for reboot NOW!
[root@host04 ~]# Connection to host04 closed by remote host
Connection to host04 closed.
```

Switch terminal windows to the other leaf node

```
[root@host05 ~]# /stage/GRID/labs/less_11/monitor.sh
```

```
root    21940 18530  0 11:01 pts/4    00:00:00 grep -i httpd -k
root    21948 18530  0 11:01 pts/4    00:00:00 grep -i httpd -k
root    21951 18530  0 11:01 pts/4    00:00:00 grep -i httpd -k
...
apache   22123 22117  0 11:01 ?        00:00:00 /usr/sbin/httpd -k
start -f /etc/httpd/conf/httpd.conf

apache   22124 22117  0 11:01 ?        00:00:00 /usr/sbin/httpd -k
start -f /etc/httpd/conf/httpd.conf

apache   22125 22117  0 11:01 ?        00:00:00 /usr/sbin/httpd -k
start -f /etc/httpd/conf/httpd.conf
...
Issue a ctl-c to stop the monitoring
```

14. Verify the `myApache` resource failover from the first leaf node to the second. Open the indicated host in a browser to verify the test page is displayed.

```
[root@host05 ~]# /u01/app/12.1.0/grid/bin/crsctl stat resource
myApache -t
--------------------------------------------------------------------
Name           Target  State        Server                State details
--------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------
myApache
      1        ONLINE  ONLINE       host05                      STABLE
--------------------------------------------------------------------

Open the indicated host in a browser to verify the test page is displayed:

http://host05.example.com

[root@host05 ~]#
```

15. After `host04` is back up, use the `crsctl relocate resource` command to move the `myApache` resource back to the original server. Verify that the `myApache` resource has been relocated back to the original node.

```
[root@host05 ~]# /u01/app/12.1.0/grid/bin/crsctl relocate resource
myApache
CRS-2673: Attempting to stop 'myApache' on 'host05'
CRS-2677: Stop of 'myApache' on 'host05' succeeded
CRS-2672: Attempting to start 'myApache' on 'host04'
CRS-2676: Start of 'myApache' on 'host04' succeeded

[root@host05 ~]# /u01/app/12.1.0/grid/bin/crsctl stat resource
myApache -t
```

```
--------------------------------------------------------------------
Name           Target  State        Server            State details
--------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------
myApache
      1        ONLINE  ONLINE       host04                   STABLE
--------------------------------------------------------------------

[root@host05 ~]#
```

16. Now, stop the `myApache` resource. When stopped, delete the resource along with the `myApache_sp` server pool. Exit the terminal session when finished.

```
[root@host05 ~]# /u01/app/12.1.0/grid/bin/crsctl stop resource
myApache
CRS-2673: Attempting to stop 'myApache' on 'host04'
CRS-2677: Stop of 'myApache' on 'host04' succeeded

[root@host05 ~]# /u01/app/12.1.0/grid/bin/crsctl delete resource
myApache

[root@host05 ~]# /u01/app/12.1.0/grid/bin/crsctl delete serverpool
myApache_sp

[root@host05 ~]# exit
logout
Connection to host05 closed.
```

17. Close all terminal windows opened for these practices.