3

Administering Oracle Clusterware

ORACLE

Objectives

After completing this lesson, you should be able to:

- Display Clusterware management proficiency
- Demonstrate Oracle Cluster Registry (OCR) backup and recovery techniques
- Manage network settings

ORACLE

Managing Oracle Clusterware

- Command-line utilities
  - `crsctl` manages clusterware-related operations:
    - Starting and stopping Oracle Clusterware
    - Enabling and disabling Oracle Clusterware daemons
    - Registering cluster resources
  - `srvctl` manages Oracle resource–related operations:
    - Starting and stopping database instances and services
- Enterprise Manager
  - Browser-based graphical user interface
  - Enterprise Manager cluster management available in:
    - Database control—within the cluster
    - Grid control—through a centralized management server

ORACLE

Managing Oracle Clusterware

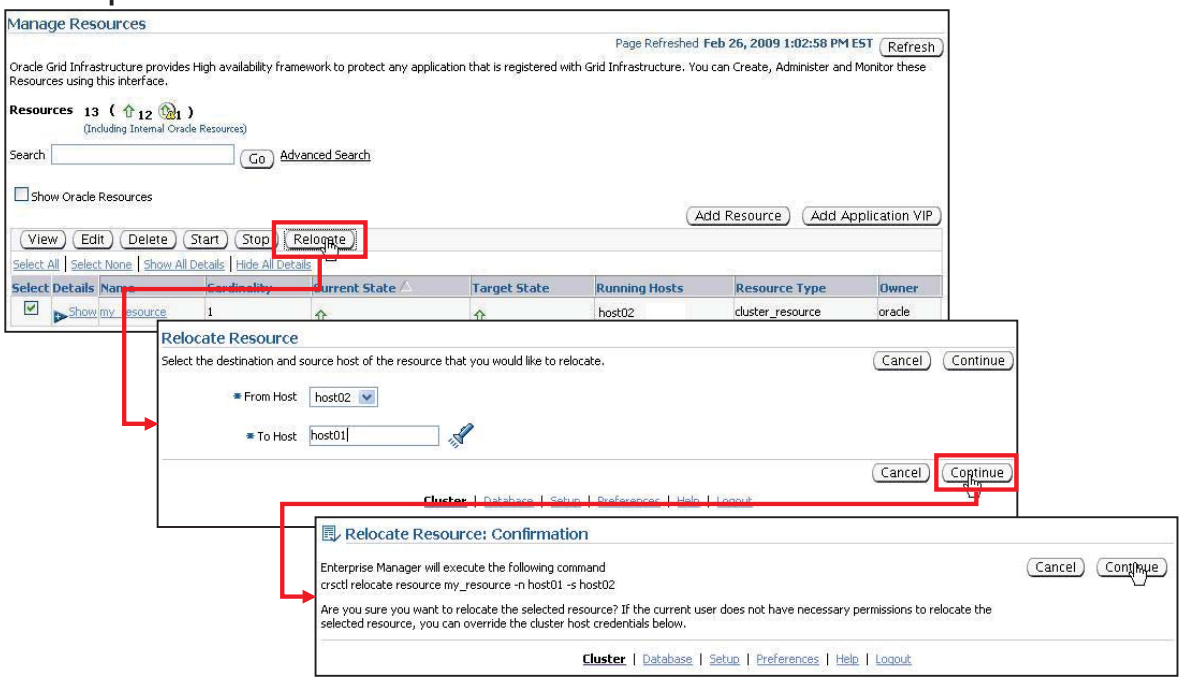
Ongoing management of Oracle Clusterware is achieved by using the `crsctl` and `srvctl` command-line utilities installed under the Oracle Grid Infrastructure home directory.

Oracle Clusterware components and resources can be monitored and managed from any node in the cluster by using `crsctl`. The `srvctl` utility provides similar monitoring and management capabilities for Oracle-related resources such as database instances and database services. Both utilities are provided with Oracle Clusterware. However, most `crsctl` commands are available only to clusterware administrators, whereas `srvctl` commands are available to other groups such as database administrators.

Enterprise Manager is Oracle's browser-based graphical management interface. Enterprise Manager can be used to manage Oracle Clusterware in two ways. Enterprise Manager Database Control can be used within a cluster to manage the functions of that cluster. Enterprise Manager Database Control requires the installation and configuration of an Oracle Database on the cluster before it can be used. Enterprise Manager Grid Control uses a centralized management server to enable the management of many clusters from a unified console.

Managing Clusterware with Enterprise Manager

Example: Relocate a resource.



Managing Clusterware with Enterprise Manager

Enterprise Manager Database Control provides facilities to manage Oracle Clusterware. This includes the ability to register and manage resources.

The example in this slide provides a typical illustration of the management interface. It shows how resources can be dynamically relocated from one node to another within your cluster. In this case, `my_resource` is relocated from `host02` to `host01`.

You can see on the Confirmation page the command that will be run to relocate the resource:

```
crsctl relocate resource my_resource -n host01 -s host02
```

Controlling Oracle Clusterware

The `crsctl` utility can be used to control Oracle Clusterware.

- To start or stop Oracle Clusterware on a specific node:

```
# crsctl start crs
```

```
# crsctl stop crs
```

- To enable or disable Oracle Clusterware on a specific node:

```
# crsctl enable crs
```

```
# crsctl disable crs
```

ORACLE

Controlling Oracle Clusterware

When a node that contains Oracle Clusterware is started, the Oracle Clusterware wrapper script is automatically started by the `/etc/init.d/ohasd` startup script. When the `crsctl` utility is used to disable Cluster Ready Services (CRS) from automatically starting, state information related to startup is placed in the `SLCS_SRC` control files, preventing automatic startup on machine reboot. To check the status of CRS, use the following syntax:

```
# 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
```

You may have to manually control the Oracle Clusterware stack while applying patches or during planned outages. You can stop Oracle Clusterware by using the `crsctl stop crs` command and start it by using the `crsctl start crs` command.

Verifying the Status of Oracle Clusterware

The `crsctl` utility can be used to verify the status of Oracle Clusterware.

- To determine the overall health on a specific node:

```
$ 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
```

- To check the viability of Cluster Synchronization Services (CSS) across nodes:

```
$ crsctl check cluster
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
```

ORACLE

Verifying the Status of Oracle Clusterware

The `crsctl` utility can be used to verify the status of Oracle Clusterware on specific nodes and across nodes. In contrast to the `crsctl` controlling commands that required the `root` access (shown in the previous slide), the `check` commands do not require `root` and may be executed by the Oracle Clusterware software owner. The overall health of the clusterware on a specific node can be obtained by using the `crsctl check crs` command. It is possible to target three of the individual daemons by using the `crsctl check <daemon>` command for the `crsd`, `evmd`, and `cssd` daemons only. These commands are processed only on the node on which they are executed. To check the viability of Cluster Synchronization Services (CSS) across all nodes, use the `crsctl check cluster` command. The output of the overall health check performed on a specific node is shown in the slide.

Determining the Location of Oracle Clusterware Configuration Files

The two primary configuration file types for Oracle Clusterware are the voting disk and the Oracle Cluster Registry (OCR).

- To determine the location of the voting disk:

```
# crsctl query css votedisk
##  STATE   File Universal Id                File Name   Disk group
--  -----   -----------------                ---------- ----------
 1. ONLINE 8c2e45d734c64f8abf9f136990f3daf8 (ASMDISK01) [DATA]
 2. ONLINE 99bc153df3b84fb4bf071d916089fd4a (ASMDISK02) [DATA]
 3. ONLINE 0b090b6b19154fc1bf5913bc70340921 (ASMDISK03) [DATA]

Located 3 voting disk(s).
```

- To determine the location of the OCR:

```
$ cat /etc/oracle/ocr.loc
ocrconfig_loc=+DATA
local_only=FALSE
```

ORACLE

Determining the Location of Oracle Clusterware Configuration Files

Oracle Clusterware uses two primary configuration file types: the voting disk and the Oracle Cluster Registry (OCR). There can be multiple redundant copies of each. You can determine the location of the voting disk by using the `crsctl query css votedisk` command on any node. This does not require the CSS daemons to be running, and the command can be executed as the Grid Infrastructure owner. The location of the OCR file can be determined by using the `cat /etc/oracle/ocr.loc` command. Because these files are always located on shared storage, the command can be executed from any node.

Note: The OCR can also be located by using the `ocrcheck` utility, provided that the path to the utility is known or the path has been added to the `PATH` environment variable.

Checking the Integrity of Oracle Clusterware Configuration Files

The following techniques are used to validate the integrity of Oracle Cluster configuration files.

- Check the `ocssd.log` for voting disks issues.

```
$ grep voting <grid_home>/log/<hostname>/cssd/ocssd.log
```

- Use the `cluvfy` utility or the `ocrcheck` command to check the integrity of the OCR.

```
$ cluvfy comp ocr -n all -verbose
```

```
$ ocrcheck
```

ORACLE

Checking the Integrity of Oracle Clusterware Configuration Files

To check the integrity of the voting disks, examine `ocssd.log`. Errors with the voting disks appear in the log. The following is a snippet of the output that indicates what an error may look like:

```
$ grep voting ocssd.log
[    CSSD]2008-09-09 10:47:09.711 [100494224] >ERROR:
   clssnmvReadFatal: voting device corrupt
   (0x00000000/0x00000000/1//dev/sda6)
[    CSSD]2008-09-09 10:47:09.711 [3082128272] >ERROR:
   clssnmvReadFatal: voting device corrupt
   (0x00000000/0x00000000/2//dev/sda7)
```

Two commands may be used to check the integrity of the OCR file. They are:

```
$ ocrcheck
$ cluvfy comp ocr -n all -verbose
```

Backing Up and Recovering the Voting Disk

- In Oracle Clusterware 11*g* Release 2, voting disk data is backed up automatically in the OCR as part of any configuration change.
- Voting disk data is automatically restored to any added voting disks.
- Using `dd` to back up and restore a voting disk ***may result in the loss of the voting disk!***
- To add or remove voting disks on non-Automatic Storage Management (ASM) storage, use the following commands:

```
# crsctl delete css votedisk path_to_voting_disk
# crsctl add css votedisk path_to_voting_disk
```

ORACLE

Backing Up the Voting Disk

In previous releases, backing up the voting disks using a `dd` command was a required postinstallation task. With Oracle Clusterware 11*g* Release 2, backing up and restoring a voting disk using the `dd` command may result in the loss of the voting disk, so this procedure is not supported.

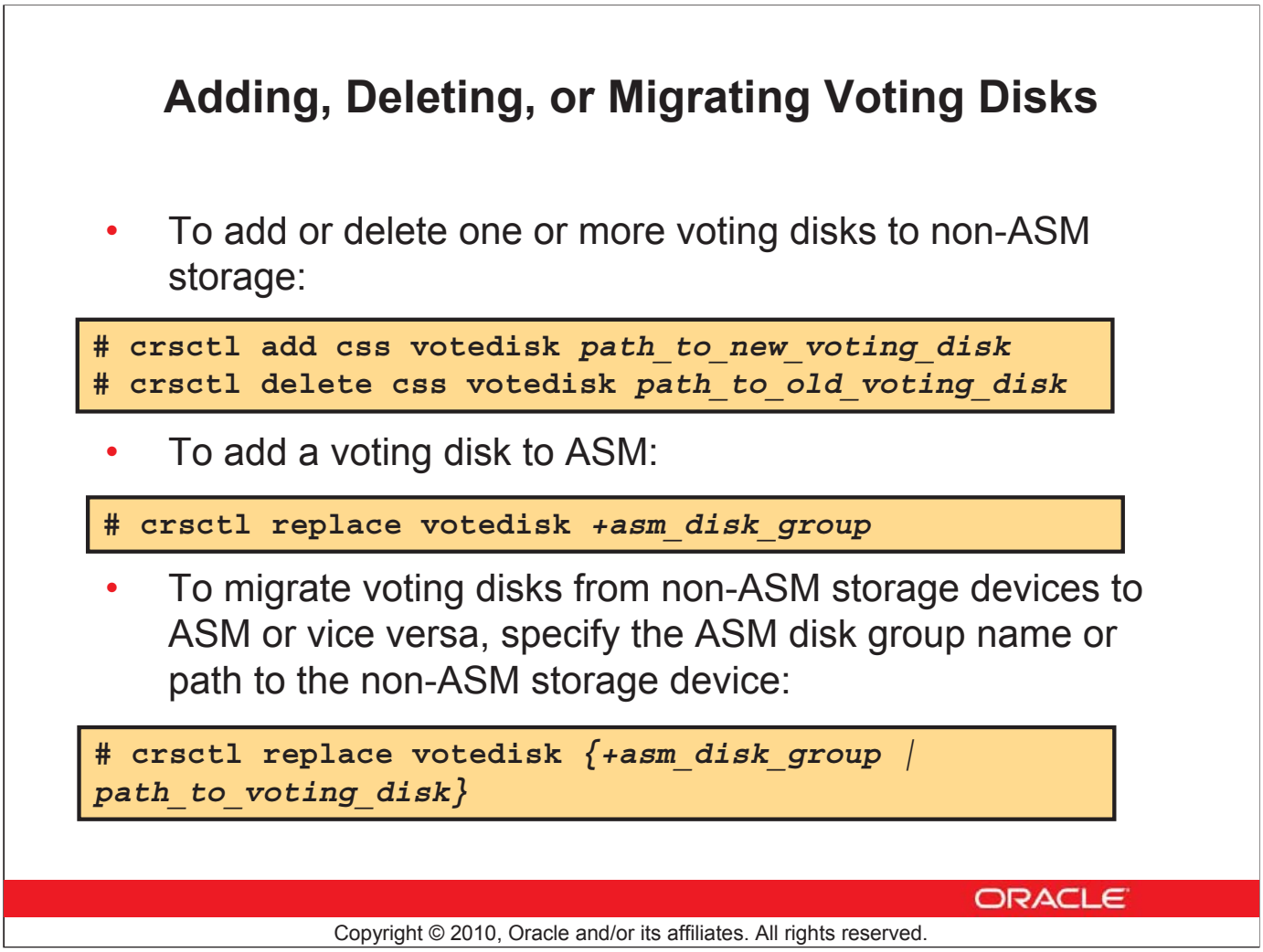
Backing up voting disks manually is no longer required because voting disk data is backed up automatically in the OCR as part of any configuration change and voting disk data is automatically restored to any added voting disks.

Recovering Voting Disks

If you have multiple voting disks on non-ASM storage, you can remove the voting disks and add them back into your environment with all the information from the other voting disks using the following commands, where *path* is the complete path of the location where the voting disk resides:

```
crsctl delete css votedisk path_to_voting_disk
crsctl add css votedisk path_to_voting_disk
```

Note: You can migrate voting disks from non-ASM storage options to ASM without taking down the cluster. To use an ASM disk group to manage the voting disks, you must set the `COMPATIBLE.ASM` attribute to 11.2.0.0.



Adding, Deleting, or Migrating Voting Disks

To add one or more voting disks to non-ASM storage, run the following command as `root`:

```
# crsctl add css votedisk path_to_voting_disk [...]
```

To add a voting disk to ASM:

```
# crsctl replace votedisk +asm_disk_group
```

To replace voting disk A with voting disk B on non-ASM storage, first add voting disk B and then delete voting disk A:

```
# crsctl add css votedisk path_to_voting_diskB
# crsctl delete css votedisk path_to_voting_diskA
```

Use the `crsctl replace votedisk` command to replace a voting disk on ASM. You do not have to delete any voting disks from ASM using this command.

To remove a voting disk, run the following command as `root`, replacing the *voting_disk_GUID* variable with one or more space-delimited, voting disk globally unique identifiers (GUIDs) you want to remove:

```
# crsctl delete css votedisk voting_disk_GUID
```

To migrate voting disks from non-ASM storage devices to ASM or vice versa, specify the ASM disk group name or path to the non-ASM storage device in the following command:

```
# crsctl replace votedisk {+asm_disk_group | path_to_voting_disk}
```

You can run this command on any node in the cluster.

Locating the OCR Automatic Backups

- The OCR is backed up automatically.
- Only one node performs the backup.
- To determine the node and location of the backup:

```
$ ocrconfig -showbackup auto
host02 2009/07/28 12:20:42  /u01/app/.../cdata/cluster01/backup00.ocr
host02 2009/07/28 08:20:41  /u01/app/.../cdata/cluster01/backup01.ocr
host02 2009/07/28 04:20:40  /u01/app/.../cdata/cluster01/backup02.ocr
host02 2009/07/27 16:20:37  /u01/app/.../cdata/cluster01/day.ocr
host02 2009/07/28 00:20:39  /u01/app/.../cdata/cluster01/week.ocr
```

- Files could be spread across nodes due to outages.
- The backup frequency and retention policies are:
  - Every four hours: CRS keeps the last three copies.
  - At the end of every day: CRS keeps the last two copies.
  - At the end of every week: CRS keeps the last two copies.

ORACLE

Locating the OCR Automatic Backups

The information contained in the OCR is much more dynamic in nature than the voting disk. Oracle Clusterware automatically performs routine backups of the OCR file. These are physical backups. Only one node has the responsibility to perform these backups, but that responsibility can transfer to any other node in the cluster when outages occur. The default target location of each automatically generated OCR backup file is the `<Grid Home>/cdata/<cluster name>` directory.

The automatic backup is on a four-hour schedule, but only a limited number of files are retained. Only the last three backups of the routine four-hour intervals are kept, with newer backups overwriting older ones. At the end of the day, a backup is taken and the last two are retained. At the end of the week, a backup is taken and the last two are retained. In conclusion, there should not be more than seven automatic backups that require storage: one four-hours old, one eight-hours old, one 12-hours old, one 24-hours old, one 48-hours old, one seven-days old, and one 14-days old. The four-hour backup interval is not based on the time of the day, but instead on an offset from the time that the clusterware was started.

The backup file names cannot be changed and are named as follows: `backup00.ocr`, `backup01.ocr`, `backup02.ocr`, `day.ocr`, `day_.ocr`, `week.ocr`, and `week_.ocr`.

Changing the Automatic OCR Backup Location

- The automatic backup location should be changed to a location shared by all nodes.

```
# ocrconfig -backuploc <path to shared CFS or NFS>
```

- The backup location will be used for both automatic and manual backups.
- It is recommended that these files be included in routine scheduled backups to an offline location.
- If CRS has been stopped on all nodes, the schedule of backups is suspended.
- On restart, a backup is not immediately taken and the backup timer is reset.

ORACLE

Changing the Automatic OCR Backup Location

Because the automatic backup is performed only by the master node to the local file system by default, it is recommended that you change the OCR automatic backup location to one that is shared by all nodes in the cluster by using the `ocrconfig -backuploc <new location>` command. This command takes one argument that is the full path of the directory name for the new location. The location will be used for both automatic and manual backups. You cannot customize the backup frequencies, the number of retained copies, or the names of the backup files. If CRS on the master node is shut down, another node becomes the master, and backups will resume on that node. If the backup location has not been changed to a common shared location, backups will exist locally across multiple nodes potentially. If CRS is stopped on all nodes during a scheduled backup, on restart, a backup will not be immediately taken and the backup timer will be reset. This could result in a longer time duration between automatic backups than the standard four-hour interval.

Because of the importance of the OCR information, it is also recommended that you manually create copies of automatically generated physical backups. You can use any backup software to copy the automatically generated backup files, and it is recommended that you do that at least once daily to a different device from where the automatic backups are.

Note: Do not place your automatic OCR backups on ASM Cluster File System (ACFS) storage.

Adding, Replacing, and Repairing OCR Locations

- Add an OCR location to either ASM or other storage device:

```
# ocrconfig -add +DATA2
# ocrconfig -add /dev/sde1
```

- To replace the current OCR location:

```
# ocrconfig -replace /dev/sde1 -replacement +DATA2
```

- To repair OCR configuration, run this command on the node on which you have stopped Oracle Clusterware:

```
[root@host03]# ocrconfig -repair -add +DATA1
```

You cannot perform this operation on a node on which Oracle Clusterware is running.

ORACLE

Adding, Replacing, and Repairing OCR Locations

You can add an OCR location after an upgrade or after completing an Oracle Grid Infrastructure installation. Oracle Clusterware can manage up to five redundant OCR locations. As `root`, run the following command to add an OCR location to either ASM or other storage device:

`# ocrconfig -add +asm_disk_group | file_name`

To replace the current OCR location using either *`destination_file`* or *`+ASM_disk_group`* to indicate the current and target OCR locations:

`# ocrconfig -replace` *`destination_file | +ASM_disk_group`* `-replacement` *`destination_file | +ASM_disk_group`*

It may be necessary to repair an OCR configuration if your configuration changes while a node is stopped. Repairing an OCR configuration involves adding, deleting, or replacing an OCR location. To repair an OCR configuration, run `ocrconfig` on the node on which you have stopped Oracle Clusterware as `root`:

`# ocrconfig -repair -add` *`file_name`* `| -delete` *`file_name`* `| -replace` *`current_file_name`* `-replacement` *`new_file_name`*

This operation changes the OCR configuration only on the node on which you run this command. For example, if the OCR location is `/dev/sde1`, use the command syntax `ocrconfig -repair -add /dev/sde1` on this node to repair its OCR configuration.

Removing an Oracle Cluster Registry Location

- To remove an OCR location, at least one other OCR must be online.
- Run the following command on any node in the cluster to remove an OCR location from either ASM or another shared location:

```
# ocrconfig -delete +DATA2
# ocrconfig -delete /dev/sde1
```

- ***Do not*** perform an OCR removal unless there is at least one other active OCR location online, or you will get an error.

ORACLE

Removing an Oracle Cluster Registry Location

To remove an OCR location, at least one other OCR must be online. You can remove an OCR location to reduce OCR-related overhead or to stop mirroring your OCR because you moved OCR to redundant storage such as RAID.

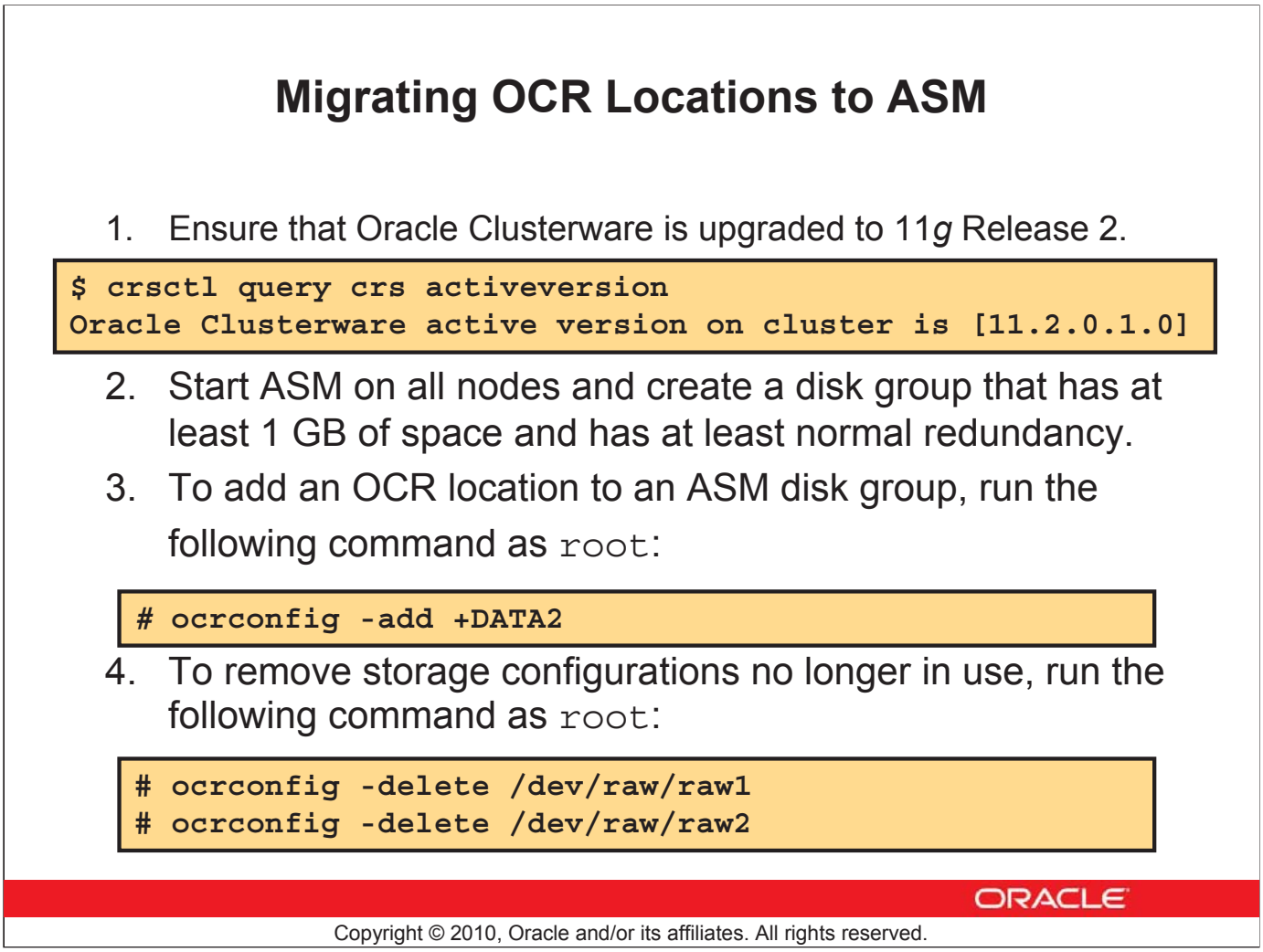
Perform the following procedure as the `root` user to remove an OCR location from your Oracle Clusterware environment:

1. Ensure that at least one OCR location other than the OCR location that you are removing is online.
2. Run the following command on any node in the cluster to remove an OCR location from either ASM or another location:

```
# ocrconfig -delete +ASM_disk_group | file_name
```

The *file_name* variable can be a device name or a file name. This command updates the OCR configuration on all the nodes on which Oracle Clusterware is running.

Caution: Do not attempt to perform an OCR removal unless there is at least one other active OCR location online otherwise you will get an error. You cannot remove the last OCR file.



Migrating OCR Locations to ASM

To improve Oracle Clusterware storage manageability, OCR is configured, by default, to use ASM in Oracle Database 11*g* Release 2. With the Oracle Clusterware storage residing in an ASM disk group, you can manage both database and clusterware storage using EM. However, if you upgrade from a previous version of Oracle Database, you can migrate your OCR location or locations to reside on ASM, and take advantage of the improvements in managing Oracle Clusterware storage. To migrate OCR locations to ASM using `ocrconfig`:

1. Ensure that Oracle Clusterware upgrade to 11*g* Release 2 is complete. Run the following command to verify the current running version:
   ```
   $ crsctl query crs activeversion
   ```
2. Use ASM Configuration Assistant (ASMCA) to configure and start ASM on all nodes in the cluster, and then create a disk group that has at least 1 GB of space and has at least normal redundancy.
3. To add an OCR location to an ASM disk group, ensure that the Clusterware stack is running and run the following command as `root`: `# ocrconfig -add +new_disk_group`
   You can run this command more than once if you add more than one OCR location.
4. To remove storage configurations no longer in use, run the following command as `root`:
   ```
   # ocrconfig -delete old_storage_location
   ```

Note: OCR inherits the redundancy of the disk group. If you want high redundancy for OCR, you must configure the disk group with high redundancy when you create it.

Migrating OCR from ASM to Other Shared Storage

1. Ensure that Oracle Clusterware is upgraded to 11*g* Release 2.

```
$ crsctl query crs activeversion
Oracle Clusterware active version on cluster is [11.2.0.1.0]
```

2. Create at least one shared file with the following permissions: **root**, **oinstall**, **640** making sure that the mount partition has at least 300 MB of space.
3. To add an OCR location, ensure that the Clusterware stack is running and run the following command as `root`:

```
# ocrconfig -add /dev/sde1
# ocrconfig -add /dev/sdf1
```

4. To remove storage configurations no longer in use, run the following command as `root`:

```
# ocrconfig -delete +DATA2
```

ORACLE

Migrating OCR from ASM to Other Shared Storage

To migrate Oracle Clusterware storage from ASM to another storage choice:

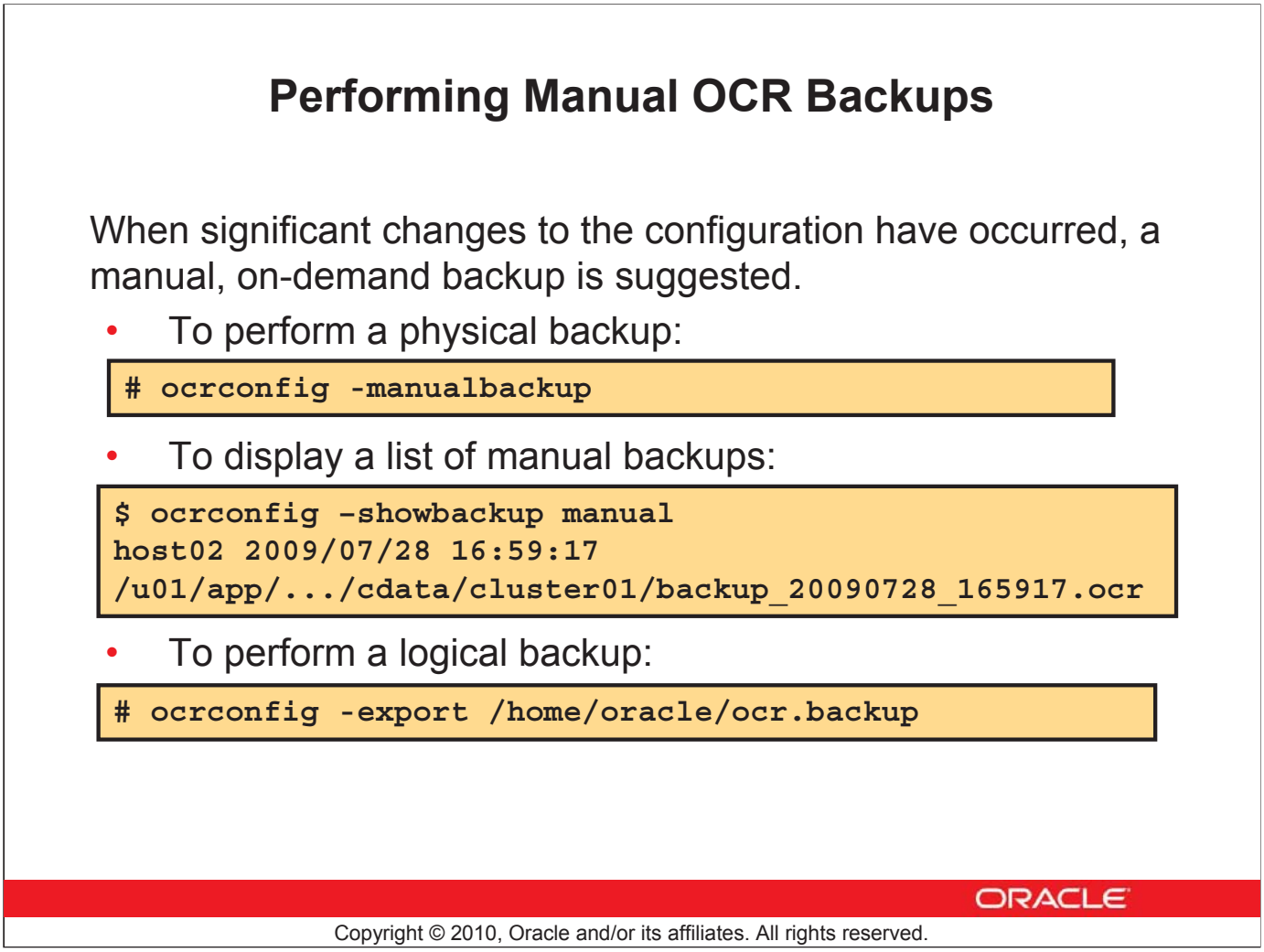
1. Ensure that Oracle Clusterware upgrade to 11*g* Release 2 is complete. Run the following command to verify the current running version: `$ crsctl query crs activeversion`.
2. Create a shared file with the following permissions: `root`, `oinstall`, `640`, making sure that the mount partition has at least 300 MB of space.
3. To add the file as an OCR location, ensure that the Oracle Clusterware stack is running and run the following command as `root`:

   `# ocrconfig -add` *`new_file_location`*

   You can run this command more than once if you add more than one OCR location.
4. To remove storage configurations no longer in use, run the following command as `root`:

   `# ocrconfig -delete` *`old_storage_location`*

   You can run this command more than once if there are multiple OCR locations configured.

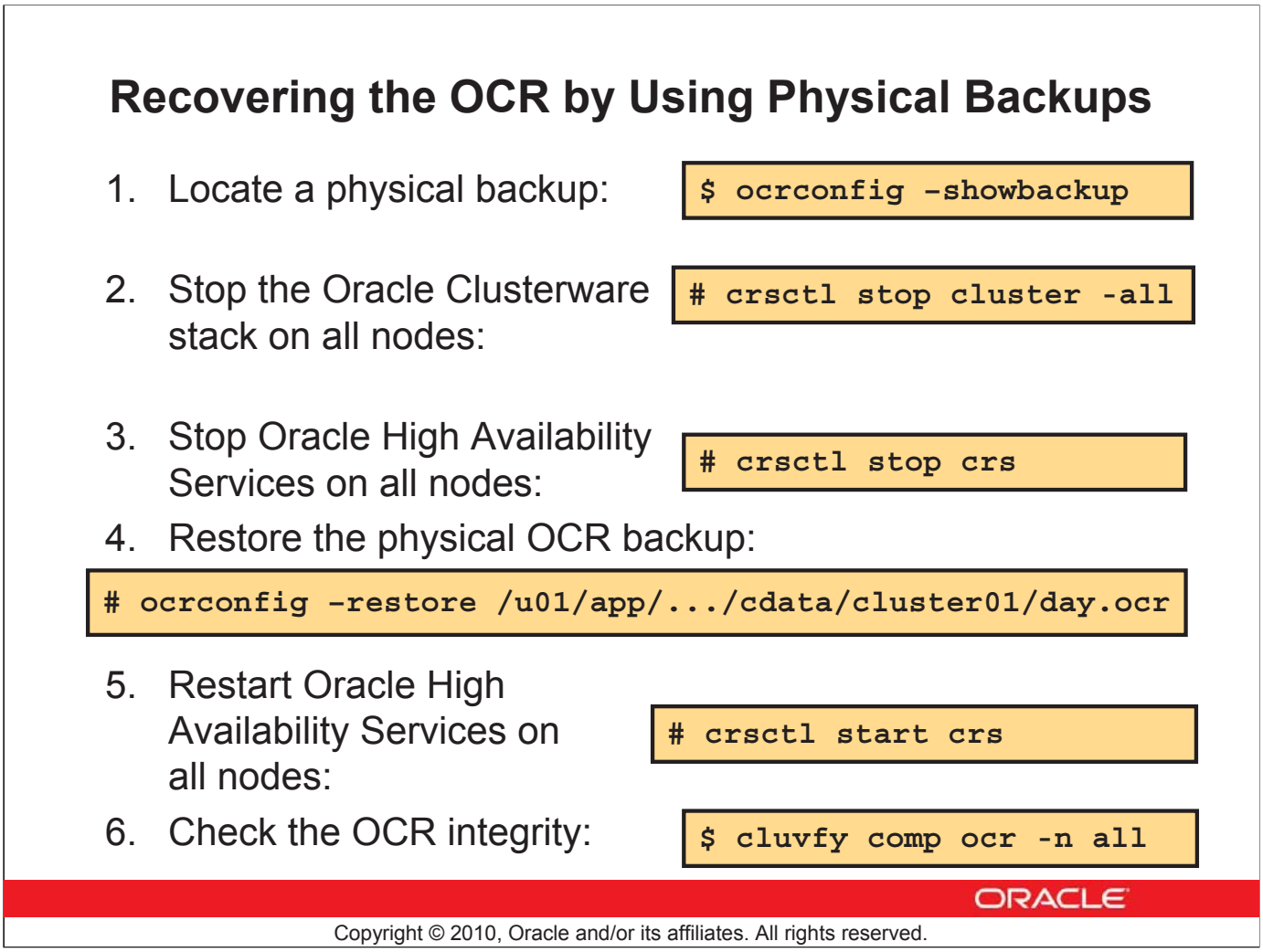


Performing Manual OCR Backups

Unlike the voting disk, the OCR content can be very dynamic in nature, especially with the High Availability framework. If a significant amount of work has been done that would cause modifications to the OCR, it is recommended that a manual backup or export be performed before the routine automatic backup occurs. This on-demand backup can be used to restore the information if the OCR becomes corrupted or lost before the automatic backup occurs.

You are not allowed to specify the name used for the manual backup. A file will be created with the name `backup_<date>_<time>.ocr` and placed into the default backup location. When a manual backup is performed, it does not affect the automatic backup interval. The `export` command will create a binary file containing a logical backup of the OCR keys and values.

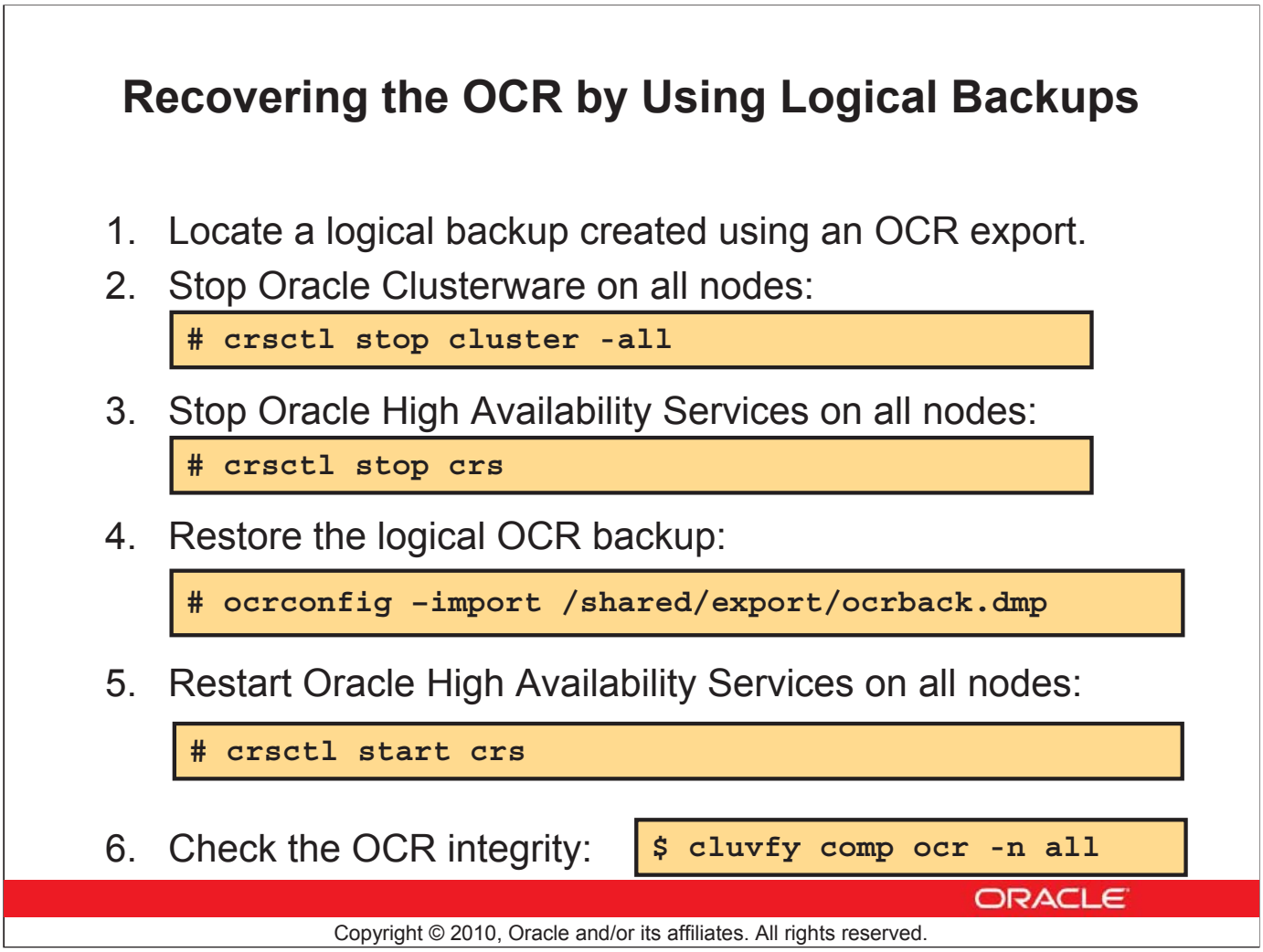
Most configuration changes that you make not only change the OCR contents but also cause file and database object creation. Some of these changes are often not restored when you restore the OCR. Do not perform an OCR restore as a correction to revert to previous configurations if some of these configuration changes fail. This may result in an OCR with contents that do not match the state of the rest of your system.



Recovering the OCR by Using Physical Backups

Use the following procedure to restore the OCR on UNIX-based systems:

1. Identify the OCR backups by using the `ocrconfig -showbackup` command. You can execute this command from any node as the `oracle` user. The output tells you on which node and which path to retrieve both automatically and manually generated backups. Use the `auto` or `manual` argument to display only one category.
2. Stop the Oracle Clusterware stack on all nodes by using the `crsctl stop cluster -all` command.
3. Stop Oracle High Availability Services on all the nodes of your cluster by executing the `crsctl stop crs` command on all the nodes as the `root` user.
4. Perform the restore by applying an OCR backup file that you identified in step 1 using the following command as the `root` user, where *`file_name`* is the name of the OCR file that you want to restore: `ocrconfig -restore` *`file_name`*.
   Ensure that the OCR devices that you specify in the OCR configuration file (`/etc/oracle/ocr.loc`) exist and that these OCR devices are valid.
5. Restart Oracle High Availability Services on the nodes in your cluster by restarting each node or running the `crsctl start crs` command as the `root` user.
6. Check the integrity of the OCR files on all nodes with the `cluvfy comp ocr -n all` command.



Recovering the OCR by Using Logical Backups

Use the following procedure to import the OCR on UNIX-based systems:

1. Identify the OCR export file that you want to import by identifying the OCR export file that you previously created using the `ocrconfig -export` *`file_name`* command.
2. Stop Oracle Clusterware on all the nodes in your RAC database by executing the `crsctl stop crs` command on the nodes as the `root` user.
3. Perform the import by applying an OCR export file that you identified in step 1 using the following command, where *`file_name`* is the name of the OCR file from which you want to import the OCR information:
   ```
   ocrconfig -import file_name
   ```
4. Restart Oracle High Availability Services on all the nodes in your cluster by restarting each node by using the `crsctl start crs` command.
5. Run the following Cluster Verify Utility (`cluvfy`) command to verify the OCR integrity, where the `-n all` argument retrieves a listing of all the cluster nodes that are configured as part of your cluster:
   ```
   cluvfy comp ocr -n all
   ```

Oracle Local Registry

- Each cluster node has a local registry for node-specific resources, called an Oracle Local Registry (OLR).
- The OLR is installed and configured when Oracle Clusterware is installed.
- One of its functions is to facilitate Clusterware startup in situations where the ASM stores the OCR and voting disks
- You can check the status of OLR by using **`ocrcheck`**:

```
$ ocrcheck -local
Status of Oracle Local Registry is as follows :
  Version                  :          3
  Total space (kbytes)     :     262120
  Used space (kbytes)      :       2204
  Available space(kbytes)  :     259916
  ID                       : 1535380044
  Device/File Name         : /u01/app/11.2.0/grid/cdata/host01.olr
       Device/File integrity check succeeded
       Local registry integrity check succeeded
       Logical corruption check succeeded
```

ORACLE

Oracle Local Registry

In Oracle Clusterware 11*g* Release 2, each node in a cluster has a local registry for node-specific resources, called an Oracle Local Registry (OLR), that is installed and configured when Oracle Clusterware installs OCR. Multiple processes on each node have simultaneous read and write access to the OLR particular to the node on which they reside, regardless of whether Oracle Clusterware is running or fully functional.

The OLR provides various Oracle Clusterware processes with access to key configuration information even when Oracle Clusterware is not running on the node. One of its functions is to facilitate the Oracle Clusterware startup process in situations where the ASM stores the OCR and voting disks. During the startup process, the OLR is referenced to determine the exact location of the voting disks. This enables the node to join the cluster. After this initial phase, ASM is started. After ASM is started, processes that require the full OCR can start and the clusterware startup process completes.

By default, OLR is located at *grid_home*`/cdata/`*hostname*`.olr`. You can manage the OLR using `ocrcheck`, `ocrdump`, and `ocrconfig` utilities with the `-local` option.

You can check the status of OLR using the `ocrcheck` utility, as follows:

```
$ ocrcheck -local
```

Oracle Local Registry (continued)

You can display the content of OLR to the text terminal that initiated the program using the OCRDUMP utility, as follows:

```
$ ocrdump -local -stdout
```

You can perform administrative tasks on OLR using the OCRCONFIG utility. To export OLR to a file:

```
$ ocrconfig –local –export file_name
```

To import a specified file to OLR:

```
$ ocrconfig –local –import file_name
```

To modify the OLR file on the local node:

```
$ ocrconfig –local –repair olr file_name
```

The olr keyword used with the -repair option is valid only when -local is used.

Determining the Current Network Settings

- To determine the list of interfaces available to the cluster:

```
$ oifcfg iflist -p -n
```

- To determine the public and private interfaces that have been configured:

```
$ oifcfg getif
eth0  192.0.2.0  global  public
eth1  192.168.1.0  global  cluster_interconnect
```

- To determine the Virtual IP (VIP) host name, VIP address, VIP subnet mask, and VIP interface name:

```
$ srvctl config nodeapps -a
VIP exists.:host01
VIP exists.:
/192.0.2.247/192.0.2.247/255.255.255.0/eth0
...
```

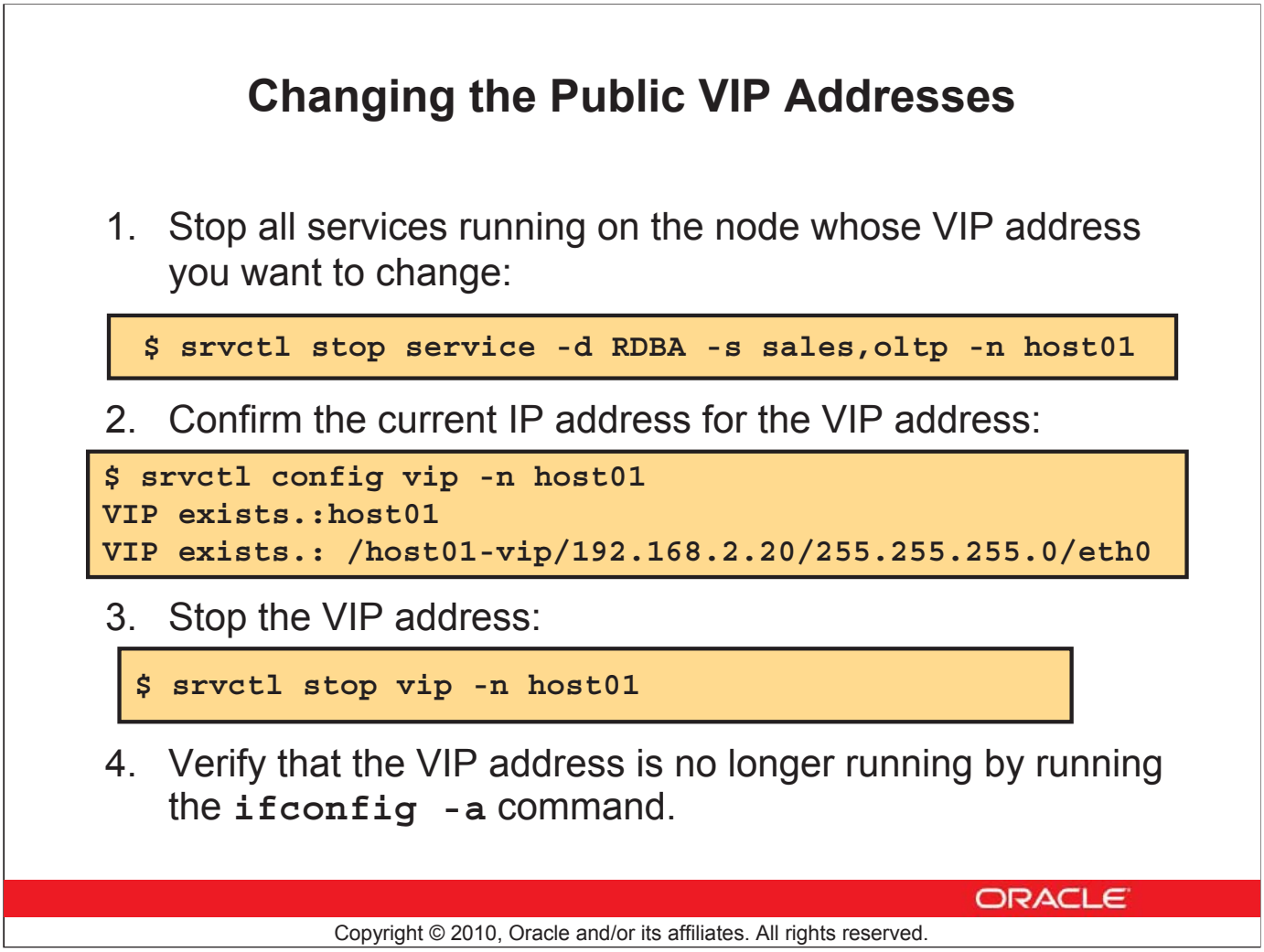
ORACLE

Determining the Current Network Settings

To determine the list of interfaces available to the cluster, use the Oracle Interface Configuration (`oifcfg`) utility. The `oifcfg iflist -p -n` command queries the operating system to find out which network interfaces are present on the node. The output lists the network number of each interface, not the IP address along with the netmask if the `-n` option is used.

To determine the public, private, and storage interfaces that have been configured for Oracle Clusterware, use the `oifcfg getif` command.

Virtual IP (VIP) addresses should be associated only with public interfaces. To determine the VIP host name, VIP address, VIP subnet mask, and VIP interface name, use the `srvctl config nodeapps -a` command.



Changing the Public VIP Addresses

Clients configured to use Public VIP addresses for Oracle Database releases before Oracle Database 11*g* Release 2 can continue to use their existing connection addresses. It is recommended that you configure clients to use single client access names (SCANs), but it is not required. When an earlier version of Oracle Database is upgraded, it is registered with the SCAN, and clients can start using the SCAN to connect to that database, or continue to use VIP addresses for connections.

If you continue to use VIP addresses for client connections, you can modify the VIP address while Oracle Database and Oracle ASM continue to run. However, you must stop services while you modify the address. When you restart the VIP address, services are also restarted on the node. Perform the following steps to change a VIP address:

1. Stop all services running on the node whose VIP address you want to change:
   `$ srvctl stop service -d db_name -s service_name_list -n my_node`
2. Confirm the current IP address for the VIP address using the `srvctl config vip` command:
   `$ srvctl config vip -n my_node`
3. Stop the VIP address using the `srvctl stop vip` command:
   `$ srvctl stop vip -n mynode`
4. Verify that the VIP address is no longer running using the `ifconfig -a` command.

Changing the Public VIP Addresses

5. Make necessary changes to the `/etc/hosts` file on all nodes and make necessary domain name server (DNS) changes to associate the new IP address with the old host name.
6. Modify node applications and provide a new VIP address:

```
# srvctl modify nodeapps -n host01 -A \
192.168.2.125/255.255.255.0/eth0
```

7. Start the node VIP.

```
# srvctl start vip -n host01
```

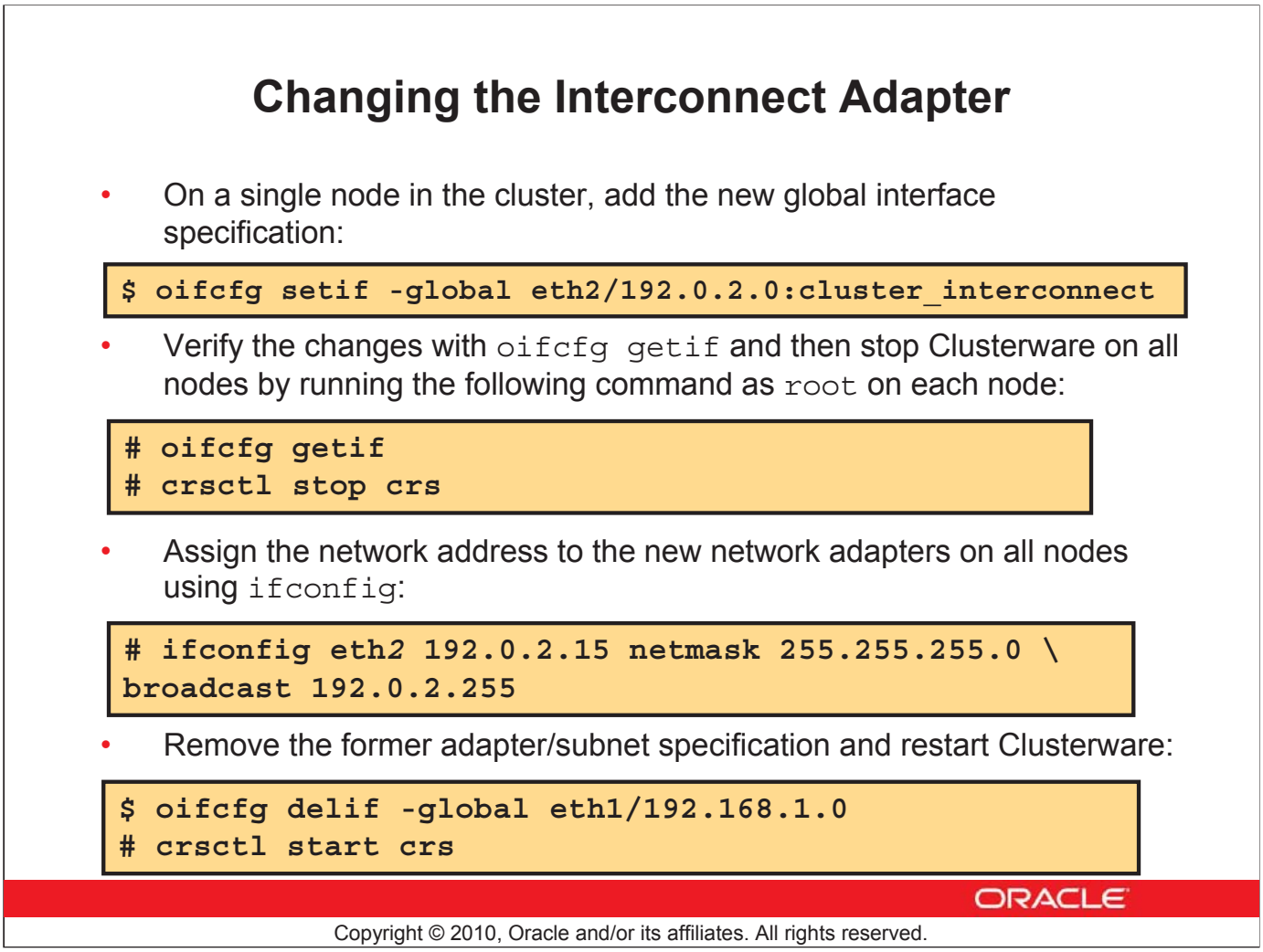
8. Repeat the steps for each node in the cluster.
9. Run `cluvfy` to verify node connectivity between all the nodes for which your cluster is configured:

```
$ cluvfy comp nodecon -n all -verbose
```

ORACLE

Changing the Public VIP Addresses (continued)

5. Make any changes necessary to the `/etc/hosts` files on all nodes and make any necessary DNS changes to associate the new IP address with the old host name.
6. Modify the node applications and provide a new VIP address using the following syntax:
   `# srvctl modify nodeapps -n` *node_name* `-A` *new_vip_address*
7. Start the node VIP by running the `srvctl start vip` command:
   `$ srvctl start vip -n mynode`
8. Repeat the steps for each node in the cluster.
   Because the `srvctl` utility is a clusterwide management tool, you can accomplish these tasks for any specific node from any node in the cluster, without logging in to each of the cluster nodes.
9. Run the following command to verify node connectivity between all the nodes for which your cluster is configured. This command discovers all the network interfaces available on the cluster nodes and verifies the connectivity between all the nodes by way of the discovered interfaces. This command also lists all the interfaces available on the nodes, which are suitable for use as VIP addresses.
   `$ cluvfy comp nodecon -n all -verbose`



Changing the Interconnect Adapter

To change the network interface for the private interconnect (for example, eth1), you must perform the change on all nodes (globally). This is because Oracle currently does not support the use of different network interface cards in the same subnet for the cluster interconnect.

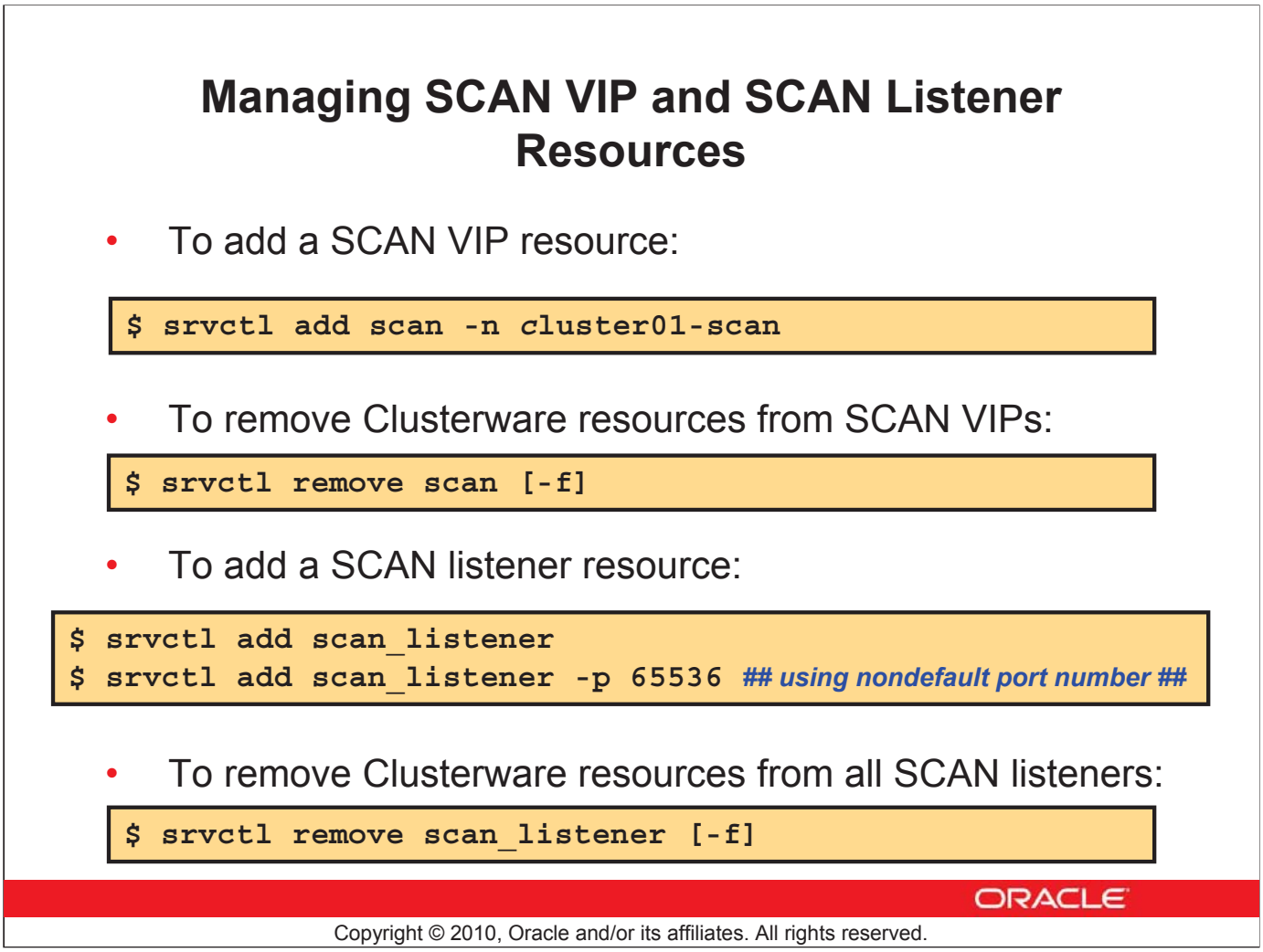
To change the network interface, perform the following steps:

1. Make sure that the Oracle Clusterware stack is up and running on all cluster nodes.
2. Use operating system commands (`ifconfig` or the command for your system) to ensure that the new or replacement interface is configured and up on all cluster member nodes.
3. On a single node in the cluster, add the new global interface specification:
   `$ oifcfg setif -global` *`interface_name/subnet`*`:cluster_interconnect`
4. On a node in the cluster, use `ifconfig` to ensure that the new IP address exists.
5. Add the new subnet, with the following command, providing the name of the interface and the subnet address. The changes take effect when Oracle Clusterware restarts:
   `$ oifcfg setif -global` *`interface_name/subnet`*`:cluster_interconnect`
6. Verify the configuration with the `oifcfg getif` command.
7. Stop Oracle Clusterware by running the following command as `root` on each node:

   ```
   # crsctl stop crs
   ```

Changing the Interconnect Adapter (continued)

8. Assign the current network address to the new network adapter by using `ifconfig`.
   As the `root` user, issue the `ifconfig` operating system command to assign the currently used private network address to the network adapter intended to be used for the interconnect. This usually requires some down time for the current interface and the new interface. See your platform-specific operating system documentation for more information about issuing the `ifconfig` command.
   You must update the operating system configuration changes because changes made using `ifconfig` are not persistent.
10. Remove the former subnet, as follows, providing the name and subnet address of the former interface: `oifcfg delif -global interface_name/subnet`
    For example: `$ oifcfg delif -global eth1/10.10.0.0`
    Note: This step should be performed only after a replacement interface is committed into the Grid Plug and Play configuration. Simple deletion of cluster interfaces without providing a valid replacement can result in invalid cluster configuration.
11. Restart Oracle Clusterware by issuing the following command as the `root` user on all nodes:
    ```
    # crsctl start crs
    ```
    You must restart Oracle Clusterware after running the `oifcfg delif` command because Oracle Clusterware, Oracle ASM, and Oracle RAC continue to use the former subnet until they are restarted.



Managing SCAN VIP and SCAN Listener Resources

The `srvctl add scan` command is used to add Oracle Clusterware resources for a given SCAN name. The command creates the same number of SCAN VIP resources as the number of IP addresses that SCAN resolves to, or three when *network_number* identifies a DHCP network and Oracle GNS configuration. Use the `srvctl add scan` command with the following syntax:

`srvctl add scan -n` *scan_name* `[-k` *network_number* `[-S` *subnet/netmask*`[/if1[|if2|...]]` where `-n` is the domain name–qualified SCAN name and `-k` is the optional network number where the SCAN IP addresses come from. Otherwise, the SCAN VIPs come from the same network as the standard VIPs. If a network number does not exist, the `-S <subnet>/<netmask>/[|if1]` clause creates the network number. The `-S` clause must be used when *<network_number>* does not exist.

To add the SCAN name `new-scan.cluster01.example.com`, run the following command:

```
# srvctl add scan -n new-scan.cluster01.example.com
```

The `srvctl add scan_listener` command can be used to add resources to the SCAN listeners. The number of SCAN listener resources created is the same as that for SCAN VIP resources. Use the `srvctl add scan_listener` command with the following syntax:

`srvctl add scan_listener [-l` *lsnr_name_prefix*`] [-p` *scan_port*`]`

where `-l` is the SCAN listener name prefix and `-p` is the SCAN listener port number.

Managing SCAN VIP and SCAN Listener Resources

- The `srvctl modify scan` command modifies the SCAN VIP configuration to match that of another SCAN VIP:

```
$ srvctl modify scan -n cluster01-scan
```

- The `srvctl modify scan_listener -u` command modifies the configuration information for all SCAN listeners to match the current SCAN VIP configuration:

```
$ srvctl modify scan_listener -u
```

ORACLE

Managing SCAN VIP and SCAN Listener Resources (continued)

The `srvctl modify scan` command modifies the SCAN VIP configuration to match that of another SCAN VIP specified with *scan_name*. If *scan_name* currently resolves to more IP addresses than when it was initially configured, new Oracle Clusterware resources for those additional IP addresses are created. If *scan_name* currently resolves to fewer IP addresses, Oracle Clusterware resources for SCAN VIP addresses with numerically higher ordinal numbers are removed until the remaining SCAN VIP resources match the number of IP addresses to which *scan_name* resolves.

Use the `srvctl modify scan` command with the following syntax:

```
srvctl modify scan [-n scan_name]
```

To modify the `cluster01-scan` SCAN VIP configuration:

```
$ srvctl modify scan -n cluster01-scan
```

The `srvctl modify scan_listener` command modifies the configuration information for all SCAN listeners. Use the `srvctl modify scan_listener` command with the following syntax:

```
srvctl modify scan_listener { -p scan_port | -u }
```

where `-p` is the new scan port and `-u` updates the SCAN listener configuration to match the current SCAN VIP configuration.

Managing SCAN VIP and SCAN Listener Resources (continued)

To change the SCAN listener port for `LISTENER_SCAN1` and update the cluster configuration, run the following command:

```
# srvctl modify scan_listener -p 1531
# srvctl modify scan_listener –u
# srvctl config scan_listener ### To verify changes ###
SCAN Listener LISTENER_SCAN1 exists. Port: TCP:1531
```

Quiz

Which of the following tools *cannot* be used to manage Clusterware operations or Oracle resources?

1. Enterprise Manager
2. `srvctl`
3. Oracle Universal Installer
4. `crsctl`

ORACLE

Answer: 3

The correct answer is 3.

Quiz

Which of the following statements regarding the Oracle Local Registry (OLR) is true?

1. Each cluster node has a local registry for node-specific resources.
2. The OLR should be manually created after installing Grid Infrastructure on each node in the cluster.
3. One of its functions is to facilitate Clusterware startup in situations where the ASM stores the OCR and voting disks.
4. You can check the status of the OLR using `ocrcheck`.

ORACLE

Answer: 1,3,4

The correct answer is 1, 3, and 4.

Summary

In this lesson, you should have learned how to:

- Display Clusterware management proficiency
- Demonstrate OCR backup and recovery techniques
- Manage network settings

ORACLE

Practice 3 Overview

This practice covers the following topics:

- Verifying, Starting, and Stopping Oracle Clusterware
- Adding and Removing Oracle Clusterware Configuration Files
- Performing a Backup of the OCR and OLR

4

Managing Oracle Clusterware

Module 4-1 Adding and Deleting Oracle Clusterware Homes

Objectives

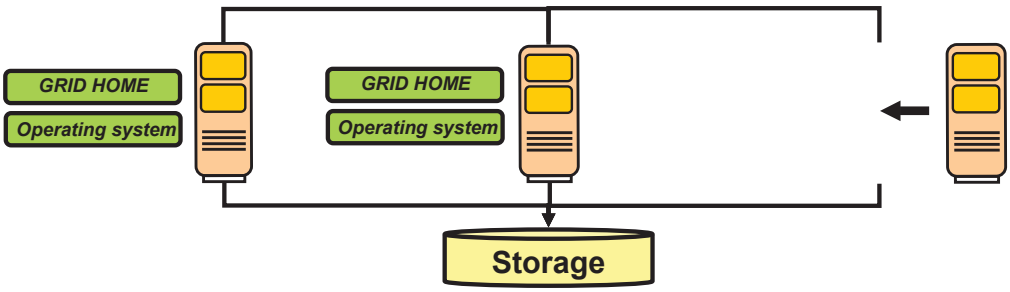
After completing this module, you should be able to:

- Perform the prerequisite steps for extending a cluster
- Use `addNode.sh` to add a node to a cluster
- Delete a node from a cluster

Adding Oracle Clusterware Homes

The `addNode.sh` shell script is used to add nodes to an existing Oracle Clusterware environment. It:

- Runs without a graphical interface
- Does not perform the prerequisite operating system tasks



ORACLE

Adding Oracle Clusterware Homes

You can use a variety of methods to add and delete nodes in an Oracle Clusterware environment:

- **Silent cloning procedures:** Copy images of an Oracle Clusterware installation to other nodes to create new clusters that have identical hardware by using the `clone.pl` script.
- **Enterprise Manager (EM) Grid Control:** Provides a GUI interface and automated wizards to the cloning procedures.
- **`addNode.sh`:** Invokes a subset of OUI functionality
- **Delete node procedure**: Removes a node from a cluster

In this module, you examine the use of the `addNode.sh` and the delete node procedure. Special attention must be given to the procedures because some steps are performed on the existing nodes, whereas other steps are performed on the nodes that are being added or removed.

Note: In this lesson, assume that `host01` and `host02` are the existing nodes, with `host03` being the node that will be added or removed.

Prerequisite Steps for Running `addNode.sh`

The following steps assume that you already have a successful Linux and Oracle Clusterware installation.

1. Make physical connections: networking, storage, and other
2. Install the operating system.
3. Perform the Oracle Clusterware installation prerequisite tasks:
   A. Check system requirements.
   B. Check network requirements.
   C. Install the required operating system packages.
   D. Set kernel parameters.
   E. Create groups and users.
   F. Create the required directories.
   G. Configure the installation owner's shell limits.
   H. Configure Secure Shell (SSH) and enable user equivalency.

ORACLE

Prerequisite Steps for Running `addNode.sh`

The `addNode.sh` script is used to extend an existing, successful Oracle Clusterware installation to more nodes. Before the `addNode.sh` script can be run, several prerequisite steps must be performed. For step 1, the new node must be physically connected to the existing cluster network infrastructure to include the public, private, storage, and other connections that may exist. Remember that all nodes must use the same adapter names for the public and private interfaces. For step 2, install a cloned image of the operating system that matches the operating system on the other nodes in the cluster, including the required service patches, drivers, and modifications to configuration files. If a cloned image is not possible, the individual modifications that were performed as prerequisite tasks for installing Oracle Clusterware will have to be performed on the new node for step 3. The provisioning of the storage prerequisite task is not listed because this step has already been performed by the existing nodes. You need to ensure that Secure Shell (SSH) is configured to operate without prompts for both the fully qualified names and nonqualified host names. This involves updates to the `authorized_keys` and `known_hosts` files of the existing nodes in addition to the new nodes being added.

Depending on the method used to transfer the operating system to the new node, some of the tasks in step 3 may have been performed and only need to be checked.

Prerequisite Steps for Running `addNode.sh`

4. Verify the installation with Cluster Verify Utility (`cluvfy`) from existing nodes.
   A. Perform a post-hardware and operating system check.

```
[grid@host01]$ cluvfy stage -post hwos -n host03
```

   B. Perform a detailed properties comparison of one existing reference node to the new node.

```
[grid@host01]$ cluvfy comp peer -refnode host01 \
-n host03 -orainv oinstall -osdba asmdba -verbose
```

ORACLE

Prerequisite Steps for Running `addNode.sh` (continued)

You have already seen a variety of methods by which Cluster Verify Utility (`cluvfy`) can be invoked. A complete reference to `cluvfy` is provided in the lesson titled "Troubleshooting Oracle Clusterware." For step 4, you invoke `cluvfy` from an existing node (`host01`) to perform a post-hardware and operating system installation check against the new node (`host03`). Next, you perform a detailed properties comparison with `cluvfy` by using an existing node as a reference node, and comparing it against the new node to be added. If errors are discovered while performing these checks, they must be corrected before continuing with the `addNode.sh` script.

In the slide examples, `host01` is an existing node and `host03` is the new node that is being added.

Adding a Node with `addNode.sh`

1. Ensure that Oracle Clusterware is successfully installed on at least one node.
2. Verify the integrity of the cluster and the node to be added (`host03`) with:

```
[grid@host01] cluvfy stage -pre nodeadd -n host03
```

3. If you are not using Grid Naming Service (GNS), run the `addNode.sh` script by using the following syntax to add `host03` to an existing cluster :

```
[grid@host01]$ cd /Grid_Home/oui/bin
[grid@host01]$ ./addNode.sh -silent \
"CLUSTER_NEW_NODES={host03}" \
"CLUSTER_NEW_VIRTUAL_HOSTNAMES={host03-vip}"
```

ORACLE

Adding a Node with `addNode.sh`

The `addNode.sh` script is used to distribute the Oracle Clusterware software from an existing node to the new nodes being added to the existing cluster. Without the silent option, the script requires that the `DISPLAY` environment variable be set; but without the silent option, no graphical windows are displayed. If you are not using GNS, for an existing cluster containing the `host01` and `host02` nodes, a new node `host03` will be added as follows:

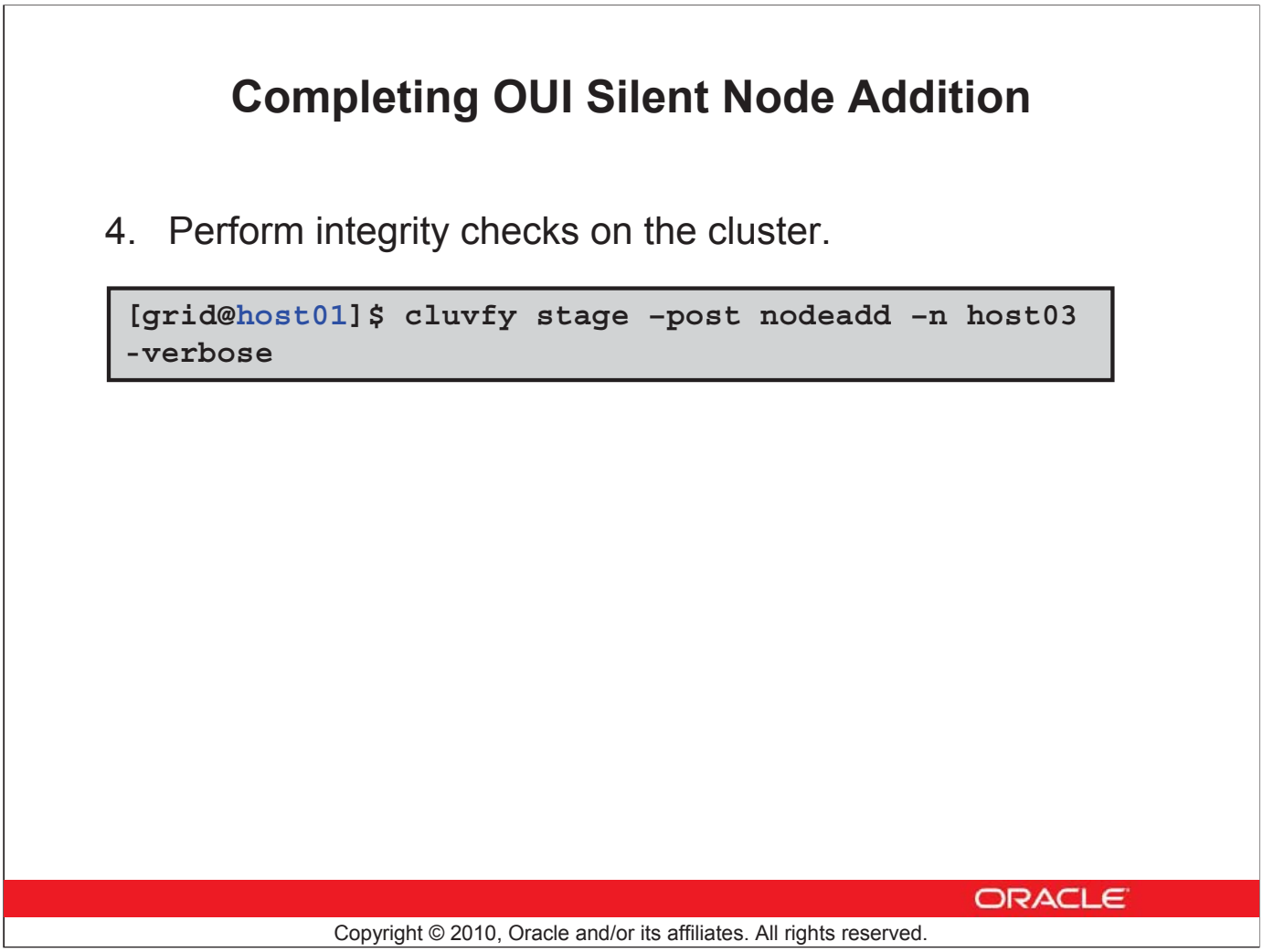
```
[grid@host01]$ ./addNode.sh -silent "CLUSTER_NEW_NODES={host03}"
"CLUSTER_NEW_VIRTUAL_HOSTNAMES={host03-vip}"
```

At the end of the `addNode.sh` script, instructions are given to run several scripts as the `root` user on selected nodes. Each script has a different name, is located in a different directory, and is run on a different node. Do not run these scripts in parallel. The instructions look like:

```
The following configuration scripts need to be executed as the root
user in each cluster node.
/u01/app/11.2.0/grid/root.sh #On nodes gr7214
```

As the `root` user, execute the scripts from a terminal window on the nodes you are adding as directed.

Note: Currently, adding a node in a GNS environment using `addNode.sh` requires `CLUSTER_NEW_VIRTUAL_HOSTNAMES` be defined.



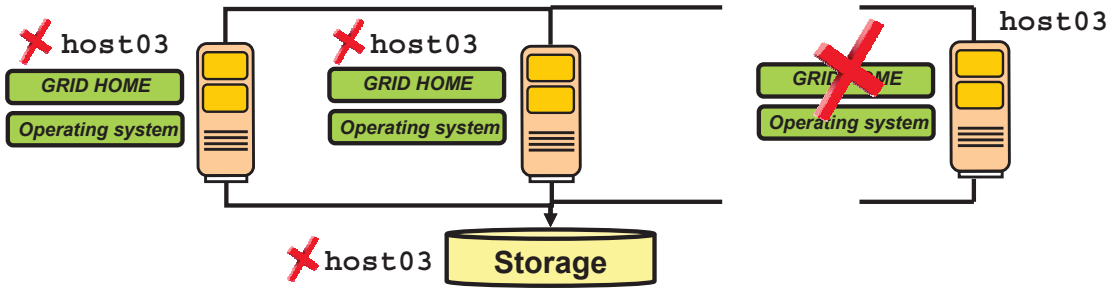
Completing OUI Silent Node Addition

For step 4, perform integrity checks on the cluster to verify a successful node addition.

Removing a Node from the Cluster

A procedure of steps is used to remove a node.

- You cannot simply remove the node from the cluster.
- Oracle Central Inventory on each node has information about all nodes.
- The Oracle Cluster Registry (OCR) contains information about all nodes.

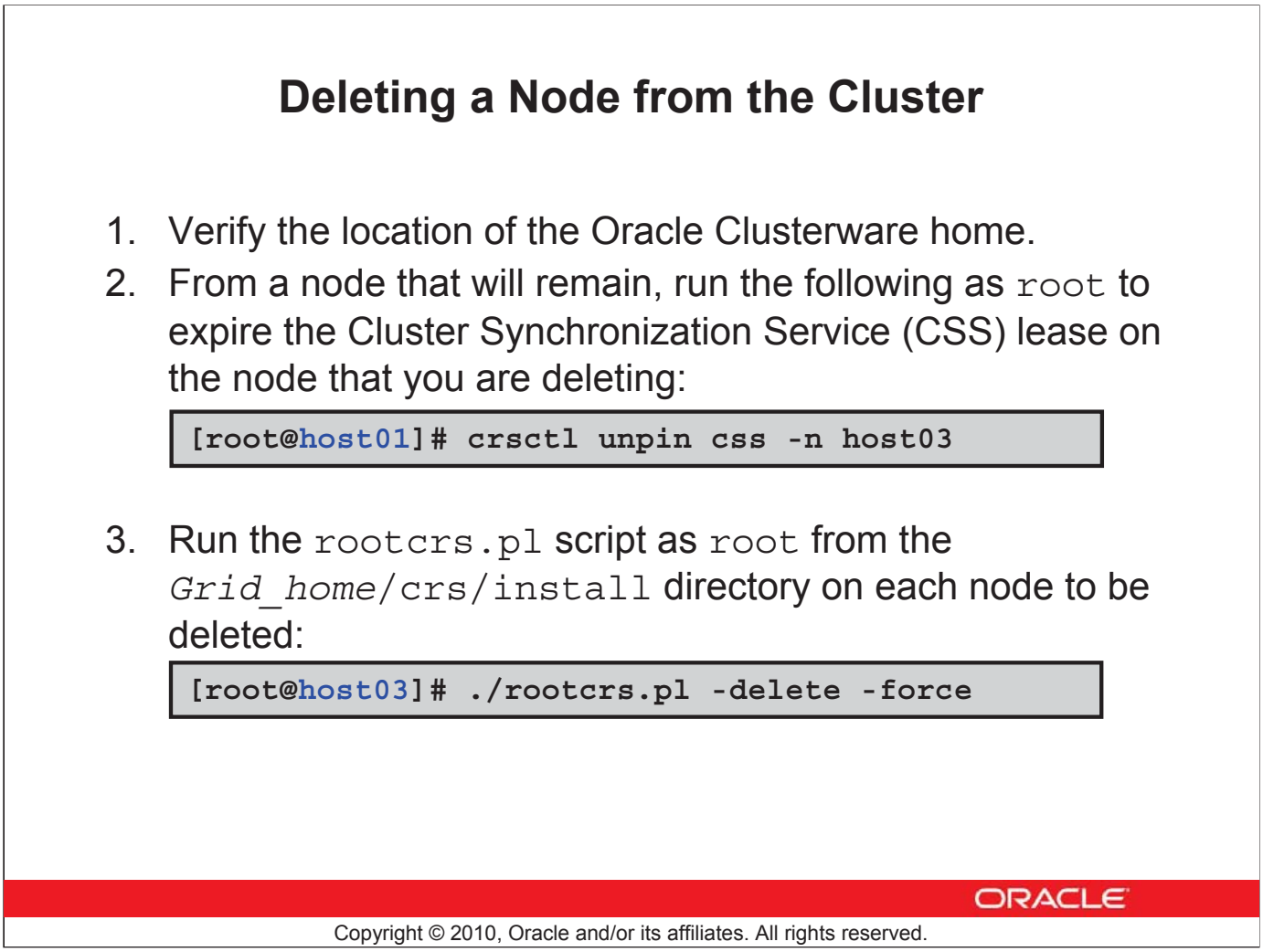


ORACLE

Removing a Node from the Cluster

On each node in the cluster, Oracle Central Inventory on that node contains information about all the nodes of the cluster. The binary OCR and voting disk also contain information about each node of the cluster. Therefore, to remove a node from the cluster properly, a procedure must be followed. You cannot simply remove the node. The procedure to remove a node from the cluster involves running a set of steps.

Deleting a node from the cluster is a multiple-step process. Some commands are run on the node to be deleted and other commands are run on an existing node of the cluster. Some commands are run by the `root` user and other commands are run by the Oracle Clusterware software owner's account. When passing arguments to a command, sometimes the existing node is passed, sometimes the node to be removed is passed, and at other times a complete list of remaining nodes is passed as an argument. This requires special attention to detail to avoid making mistakes during the process.



Deleting a Node from the Cluster

If you run a dynamic Grid Plug and Play (GPnP) cluster using Dynamic Host Configuration Protocol (DHCP) and GNS, skip to step 3.

Step 1: Verify the location of the Oracle Clusterware home. This directory should be consistent on all nodes.

Step 2: Expire the CSS lease on the node you are deleting. The `crsctl unpin` command will fail if Cluster Synchronization Services (CSS) is not running on the node being deleted. Run the `olsnodes -s -t` command to show whether the node is active or pinned. If the node is not pinned, go to step 3.

Note: You cannot unpin a node that has a Real Application Clusters (RAC) instance older than 11.2.0 on that node, if you installed Oracle Clusterware 11*g* release 2 (11.2) on that node.

Step 3: Disable the Oracle Clusterware applications and daemons running on the node. Then if you run a dynamic Grid Plug and Play cluster using DHCP and GNS, continue to step 4.

Note: This procedure assumes that the node to be removed can be accessed. If you cannot execute commands on the node to be removed, you must manually stop and remove the VIP resource using the following commands as `root` from any node that you are not deleting:

```
# srvctl stop vip -i vip_name -f
# srvctl remove vip -i vip_name -f
```

where `vip_name` is the Virtual IP (VIP) for the node to be deleted.

Deleting a Node from the Cluster

4. From a node that will remain, delete the node from the cluster with the following command run as `root`:

```
[root@host01]# crsctl delete node -n host03
```

5. On the node to be deleted, as the user that installed Oracle Clusterware, run the following command from the *Grid_home*`/oui/bin` directory:

```
[grid@host03]$ ./runInstaller -updateNodeList
ORACLE_HOME=Grid_Home "CLUSTER_NODES=
{host03}" CRS=TRUE -local
```

ORACLE

Deleting a Node from the Cluster (continued)

Step 4: As the `root` user, delete the node from the cluster from a node that will remain in the cluster. Then if you run a dynamic Grid Plug and Play cluster using DHCP and GNS, skip to step 7.

Step 5: As the user that installed Oracle Clusterware on the node to be deleted, run the command as follows:

```
[grid@host03]$ cd $Grid_Home/oui/bin
[grid@host03]$ ./runInstaller -updateNodeList \
ORACLE_HOME=Grid_Home "CLUSTER_NODES={host03}" CRS=TRUE -
local
```

Deleting a Node from the Cluster

6. On the node that you are deleting, run the `runInstaller` command as the user that installed Oracle Clusterware.
   A. If you have a shared home:

```
[grid@host03]$ ./runInstaller -detachHome
ORACLE_HOME=/u01/app/11.2.0/grid
```

   B. For a nonshared home, deinstall the Oracle Clusterware home:

```
[grid@host03]# ./deinstall -local
```

7. On any remaining node, as the Grid software owner, update the node list:

```
[grid@host01]$ cd /Grid_home/oui/bin
[grid@host01]$ ./runInstaller -updateNodeList \
ORACLE_HOME=/u01/app/11.2.0/grid \
"CLUSTER_NODES={host01,host02}" CRS=TRUE
```

ORACLE

Deleting a Node from the Cluster (continued)

Step 6:

A. For a shared home from the *Grid_home*/oui/bin directory as the clusterware owner:

```
[grid@host03]$ ./runInstaller -detachHome
ORACLE_HOME=/u01/app/11.2.0/grid
```

B. For a nonshared home from the *Grid_home*/deinstall directory as the Grid software owner:

```
[grid@host03]# ./deinstall -local
```

Warning: `deinstall -local` can damage the cluster installation.

Step 7: On any remaining node, update the node list. Run this command as the Grid software owner:

```
[grid@host01]$ /Grid_Home/oui/bin/runInstaller \
-updateNodeList ORACLE_HOME=/u01/app/grid \
"CLUSTER_NODES={host01,host02}" CRS=TRUE
```

Deleting a Node from a Cluster (GNS in Use)

If your cluster uses GNS, do the following (steps from the previous procedure):

3. Run the `rootcrs.pl` script as `root` from the *`Grid_home`*`/crs/install` directory on each node to be deleted.
4. From a node that will remain, delete the node from the cluster.
7. On any remaining node as the Grid software owner, update the node list.

ORACLE

Deleting a Node from a Cluster (GNS in Use)

If your cluster is a dynamic Grid Plug and Play cluster using DHCP and GNS, the procedure for deleting a node from the cluster is simplified. Run just steps 3, 4, and 7 from the previous procedure.

Deleting a Node from the Cluster

8. On any remaining node, verify that the specified nodes have been deleted from the cluster.

```
[grid@host01]$ cluvfy stage -post nodedel -n
host03 [-verbose]
```

ORACLE

Deleting a Node from the Cluster (continued)

For step 8, run the `cluvfy` command to verify that specific nodes have been successfully deleted from a cluster. Typically, this command verifies that the node-specific interface configuration details have been removed, the nodes are no longer a part of the cluster configuration, and proper Oracle Automatic Storage Management (ASM) cleanup has been performed.

Quiz

The `addNode.sh` script can generate fixup scripts to correct prequisites for new nodes for an existing cluster.

1. True
2. False

ORACLE

Answer: 2

No, `addNode.sh` requires that the new node to be added be already configured properly. The `cluvfy` utility can be used to generate fixup scripts.

Summary

In this module, you should have learned how to:

- Perform the prerequisite steps for extending a cluster
- Use `addNode.sh` to add a node to a cluster
- Delete a node from a cluster

ORACLE

Module 4-2 Patching Oracle Clusterware

Objectives

After completing this module, you should be able to:

- Describe the types of patches and upgrades available
- Design for rolling patches and rolling upgrades
- Compare software versions with the active version
- Install a patchset with the Oracle Universal Installer (OUI) utility
- Install a patch with the `opatch` utility

ORACLE

Out-of-Place Oracle Clusterware Upgrade

- Oracle Clusterware 11*g* Release 2 supports only out-of-place upgrades.
  - An in-place upgrade is installed in the existing Clusterware home and replaces the older software.
  - An out-of-place upgrade has both Clusterware versions present on the nodes at the same time, in different Grid homes; but only one is active.
- Install Oracle Clusterware in a separate home before the upgrade reduces the down time for cluster upgrades.
- The active software version and Grid home location are stored in the OCR.
- There is a clusterwide active version that reflects the version of the communication protocols and shared disk data structures that are currently in use on the cluster.

ORACLE

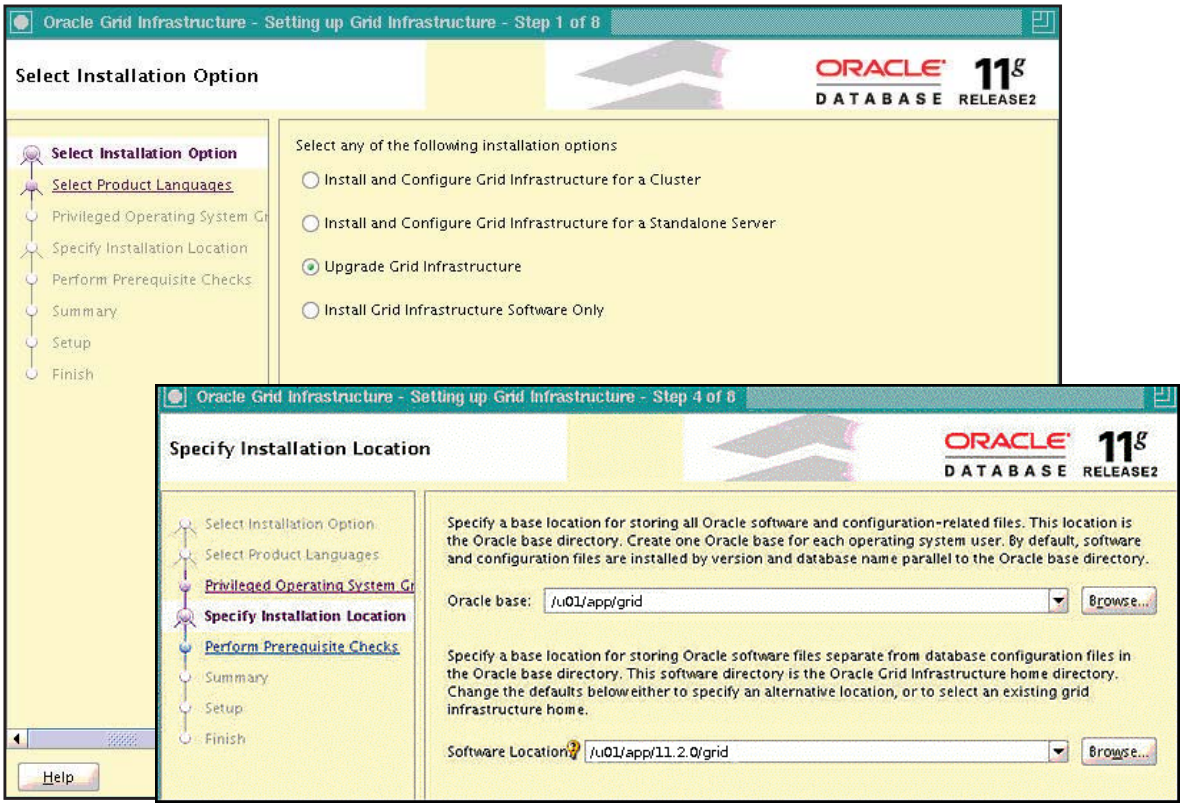
Out-of-Place Oracle Clusterware Upgrade

Oracle Clusterware 11*g* Release 2 supports only out-of-place upgrades. Out-of-place upgrades mean the new software is placed in a different Oracle Clusterware home directory. In past versions, in-place upgrades were performed. In-place upgrades (or downgrades) mean that the software being installed is placed in the same home directory as the existing or active software. With the out-of-place upgrade, both versions of the software are present on the nodes at the same time, in different Grid homes; but only one is active.

Installing Oracle Clusterware in a separate home before the upgrade reduces the planned outage time required for cluster upgrades, which assists in meeting availability service-level agreements. After the software is installed, you can upgrade the cluster by stopping the previous version of the software and starting the new version node by node (known as a rolling upgrade).

Each node maintains a “software version” that indicates the version of the Oracle Clusterware that is active on the machine. The software version is stored in the OCR. There is also a clusterwide “active version” for the entire cluster that reflects the version of the communication protocols and shared disk data structures that are currently in use on the cluster.

Oracle Clusterware Upgrade



Oracle Clusterware Upgrade

Upgrading Oracle Clusterware using the Oracle Database 11*g* Release 2 OUI allows you to perform an out-of-place upgrade. The Specify Installation Location page is displayed. The Specify Location field is not read-only, as you can see in the graphic in the slide. All you can enter is a new directory under the current Oracle base in which to install the new Clusterware software, performing an out-of-place upgrade.

Types of Patches

- CRS Patchset:
  - Is installed with the OUI utility
  - Has the naming convention: 10.2.0.4.0,11.1.0.7.0,11.2.0.1
  - Is cumulative for each base release
- CRS Bundle Patch (BP):
  - Is installed with the `opatch` utility
  - Has the naming convention: 11.1.0.6.0 BP#1, BP#2, BP#3
  - Is a fully regression tested bundle of around 20–30 patches
  - Is cumulative for each BP on the same patchset level
- CRS Merge Label Request (MLR) or "One Off" patch:
  - Is installed with the `opatch` utility
  - Generally contains priority 1 fixes not in a bundle yet
- Patch Set Update (PSU)
  - Is a cumulative patch containing recommended bug fixes released on a quarterly schedule
  - Installed with the `opatch` utility

ORACLE

Types of Patches

There are different types of patches that are available for Oracle Clusterware. The first type is a patchset. This is a set of patches provided on top of a base release (for example, 10.2.0.1.0 or 11.1.0.2.0). A patchset increments the fourth digit of the release number. A patchset will include updates for Oracle Clusterware and updates for the Oracle RDBMS software. You must update Oracle Clusterware before updating the Oracle RDBMS product. Though the reverse is not true, Oracle Clusterware may be patched without patching the RDBMS. Patchsets are released less frequently than other types of patches. They are cumulative and may contain hundreds of fixes. The Oracle Universal Installer (OUI) utility is always used to install patchsets.

The second common type of patch available for Oracle Clusterware is known as a CRS Bundle Patch (BP). A CRS BP is a small grouping of individual fixes, usually around 20 to 30, that is fully regression tested and released more frequently than patchsets. Each CRS bundle patch on the same patchset level is cumulative for that patchset level only. The naming convention is 11.1.0.6.0 BP#1, 11.1.0.6.0 BP#2, and so on. When the patchset level is incremented, the CRS bundle patch numbering will restart. CRS bundle patches are always applied with the `opatch` utility instead of the OUI utility. BPs are binaries and do not require relinking.

Another type of patch that may be applied to Oracle Clusterware is known as a CRS Merge Label Request or a "One Off" patch. These are installed with the `opatch` utility and are usually priority 1 fixes that are not contained in a bundle yet. One MLR can contain multiple fixes.

Types of Patches (Continued)

Patch Set Updates (PSU) are patches containing important updates and fixes accumulated from the last PSU released. The purpose of these types of patches is to provide important updates and fixes in a single, well tested package on a regular schedule. The PSUs generally supersede older bundle patches and establish a new baseline version for easier tracking.

Patch Properties

- All upgrades and patchsets are installed as rolling.
- All patch bundles can use the minimum down-time procedures. Most patch bundles are rolling upgradeable.
- Individual patches may be rolling; check the patch.
- After 11.2.0.2, patch bundles can be in-place or out-of-place.

Patch type	Tool	Method	Rolling Upgradable
Patchset	OUI	Out-of-place	Yes
Patch bundle	OPatch/Enterprise Manager	In-place	Most (check)
One-off patch	OPatch/Enterprise Manager	In-place	Most (check)

ORACLE

Patch Properties

Depending on their type, patches can have different methods and utilities for installation.

Patchsets can be installed as a rolling patch, with the Oracle Universal Installer (OUI). Patchsets are installed out-of-place.

Patch bundles can be installed with either Enterprise Manager (EM) or OPatch. Even EM Database Control can be used to patch Oracle Clusterware and the RDBMS software. Patch bundles are installed in-place and most bundles can be installed as a rolling patch, but check the `README.txt` file that comes with the patch, or the patch metadata.

One-off patches can only be installed in-place with EM or OPatch. Just like patch bundles, most one-off patches can be installed as a rolling patch, if they were created to be rolling. If the patch or patch bundle were not labeled as rolling, it cannot be used in a rolling manner.

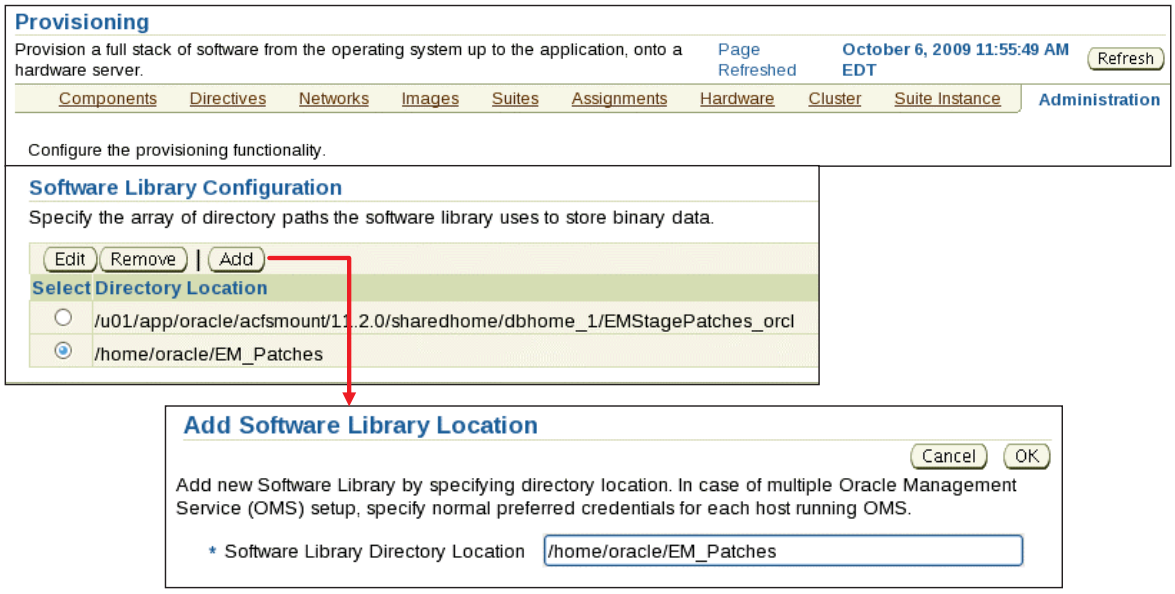
Any one-off patch or patch bundle may use the minimum down-time method.

The use of Enterprise Manager to install patches is covered in detail in the *Oracle Database 2 Day + Real Application Clusters Guide 11g Release 2*.

Configuring the Software Library

Use the Provisioning page of Enterprise Manager.

- Add a software library location:



ORACLE

Configuring the Software Library

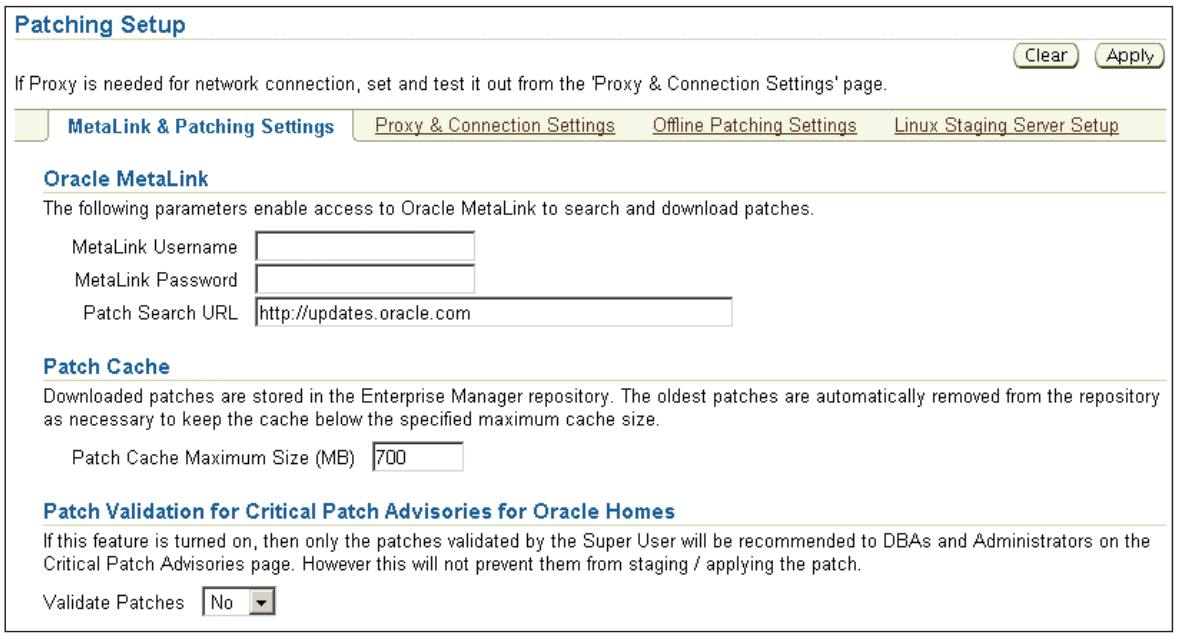
To use the patching features of Enterprise Manager, you configure a software library location. You can have one or more library locations. This will be the directory that patches will be stored in when they are transferred to the local cluster.

To navigate to the Provisioning page from the Database home page:

1. Click the Software and Support tab.
2. In the Deployment Procedure Manager section, click the Deployment and Provisioning Software Library link.

On the Provisioning page, you will find the Software Library Configuration at the bottom of the page. The software library directory location must reference an existing directory.

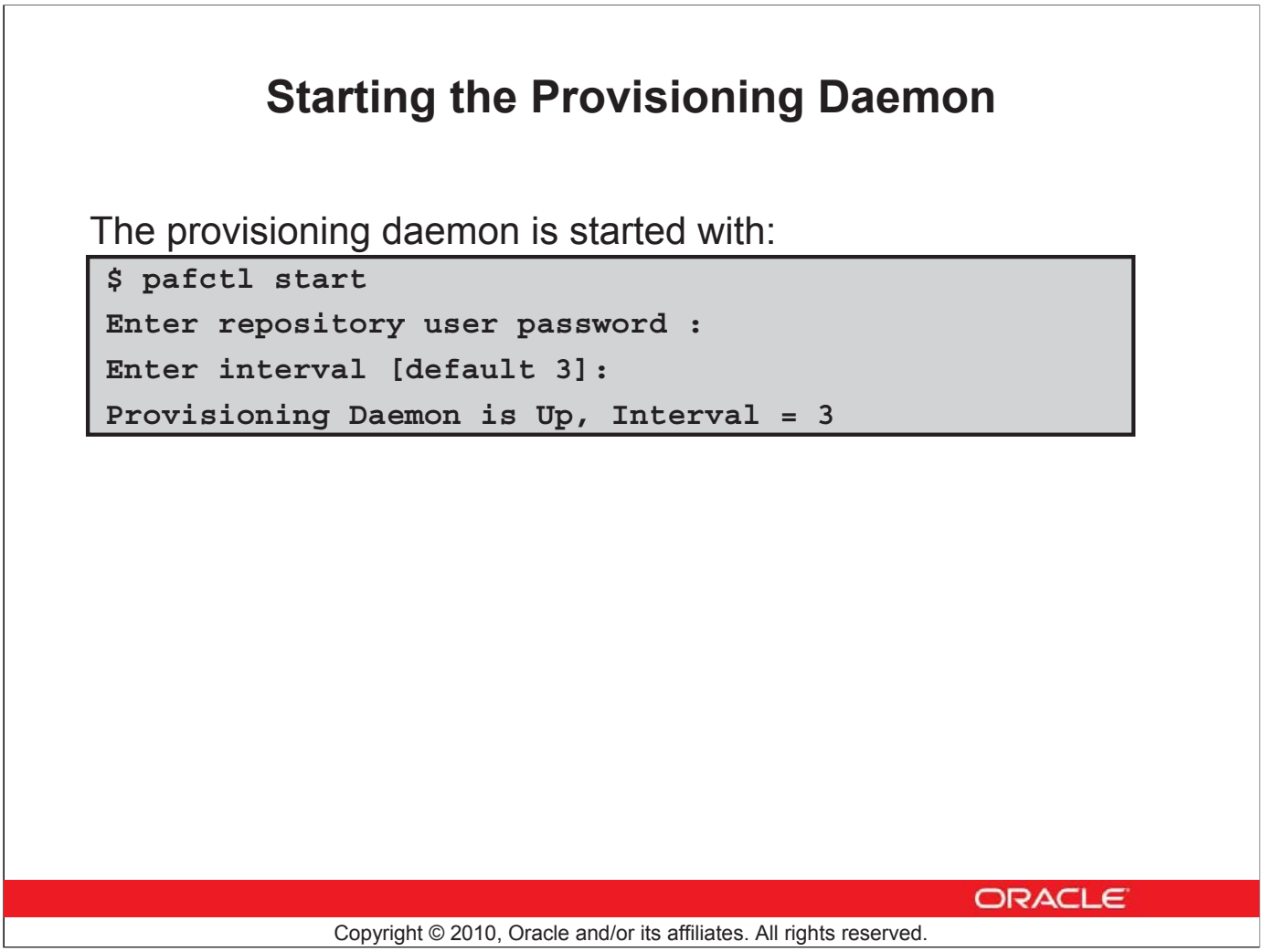
Setting Up Patching



Setting Up Patching

Patching is simplified if the Enterprise Manager is configured to connect to My Oracle Support (formerly, MetaLink). Available patches can be downloaded and deployed through Enterprise Manager. The Patching Setup page can be accessed by a superuser, using the setup button. The patching configuration can also be supplied during installation.

To complete the patching configuration, you must supply the credentials to connect to My Oracle Support. This assumes that your cluster has access to the Patch Search URL: http:/updates.oracle.com either directly or through a proxy. The Proxy and Connections Settings tab allows you to configure the Internet connection.



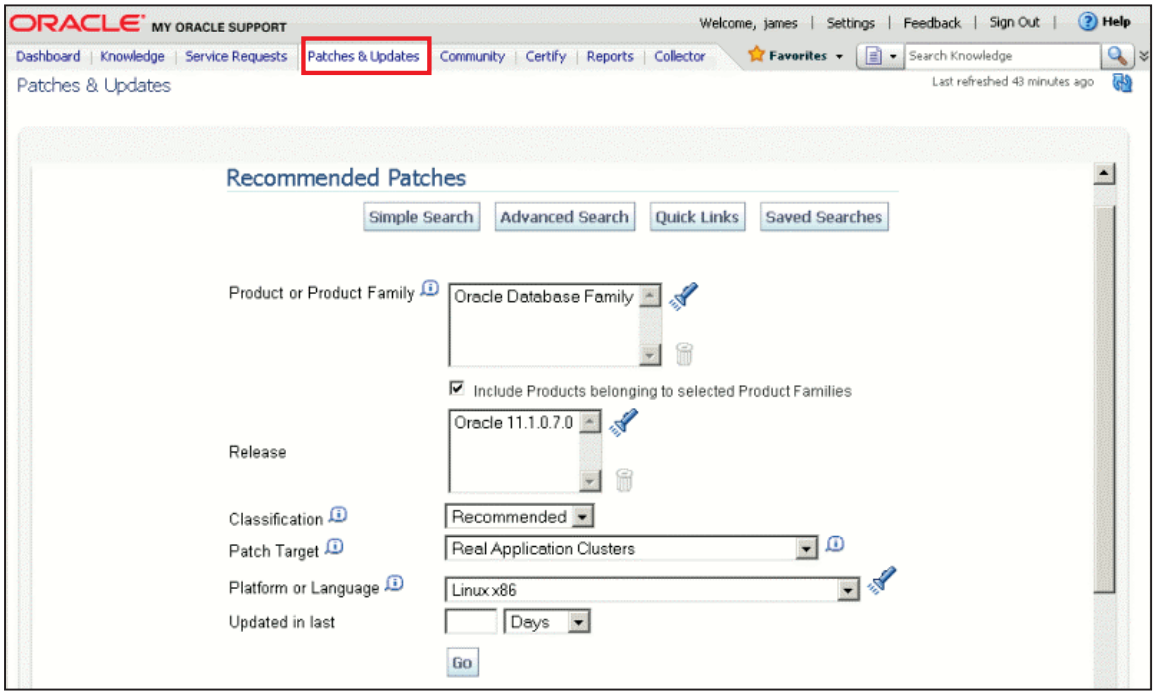
Starting the Provisioning Daemon

The Deployment Procedure Manager allows you to view, edit, run, and monitor deployment procedures. Deployment procedures are best practices provided by Oracle for various provisioning tasks.

In order for the Deployment Procedure Manager to function properly, the Provisioning Daemon job should be running. The provisioning daemon monitors the status of running deployment procedures.

The `pafctl` command allows you to start, stop, and view the status of the provisioning daemon.

Obtaining Oracle Clusterware Patches



ORACLE

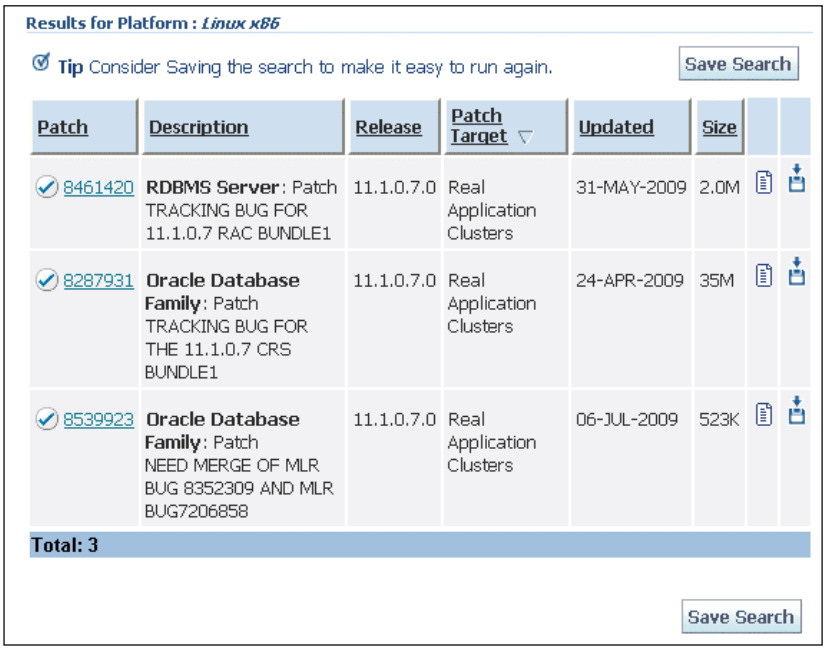
Obtaining Oracle Clusterware Patches

The latest patchsets and recommended CRS BPs can be downloaded from the My Oracle Support Web site at the following URL:

http://support.oracle.com/

After signing in to the Web site, click the "Patches & Updates" tab. For the latest patchset, click the "Quick Links to: Latest Patchsets, Mini Packs and Maintenance Packs" link. To locate the latest CRS BP, if there are any available, start from the "Patches & Updates" tab and click the "Oracle Database" link under the Recommended Patches heading. The Recommended Patches Search page will then be displayed. Leave the Product field set to the value of "Oracle Database Family," change the value for the Release field, change the value of the Patch Target field to "Real Application Clusters," change the value of the Platform field, and then click Go. The latest CRS BP will be listed if available.

Obtaining Oracle Clusterware Patches



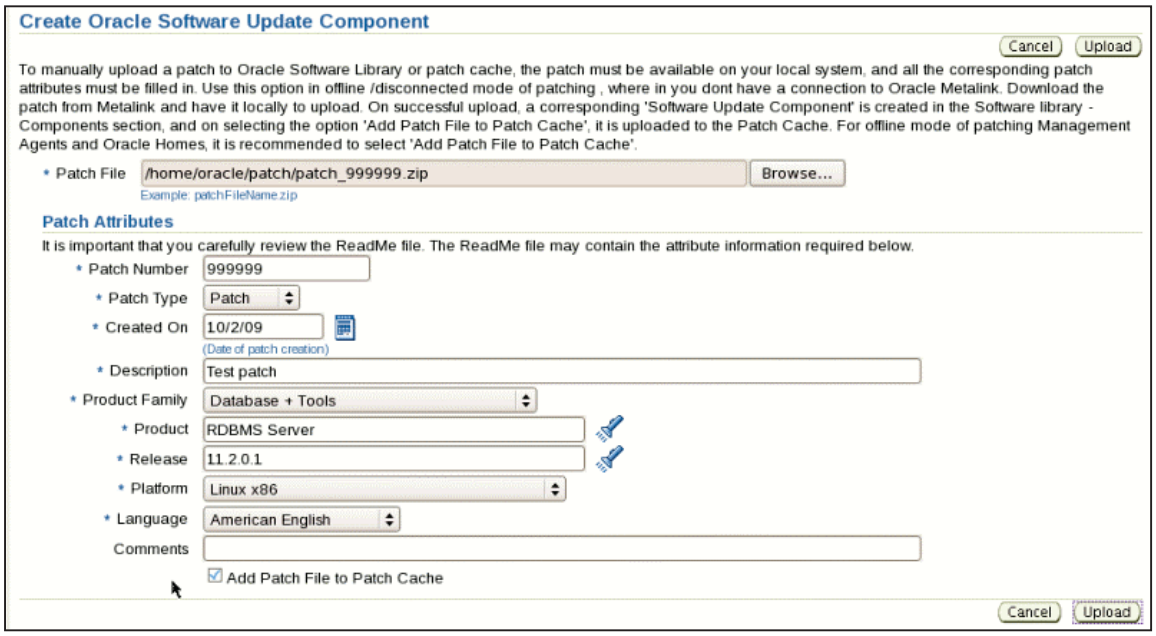
ORACLE

Obtaining Oracle Clusterware Patches (continued)

Individual patches that are not included in a BP are usually made available as a result of a service request logged with Oracle Support for a specific problem.

Note: This slide queried for the 11.1.0.7 release to show an example for a BP. At the time of writing this course, there were no BPs available for 11.2.0.

Uploading Patches



ORACLE

Uploading Patches

When a direct connect to the patch URL is not available, patches can be downloaded to a local machine and then uploaded to the Software Library. To upload patches to the Software Library using Enterprise Manager database control:

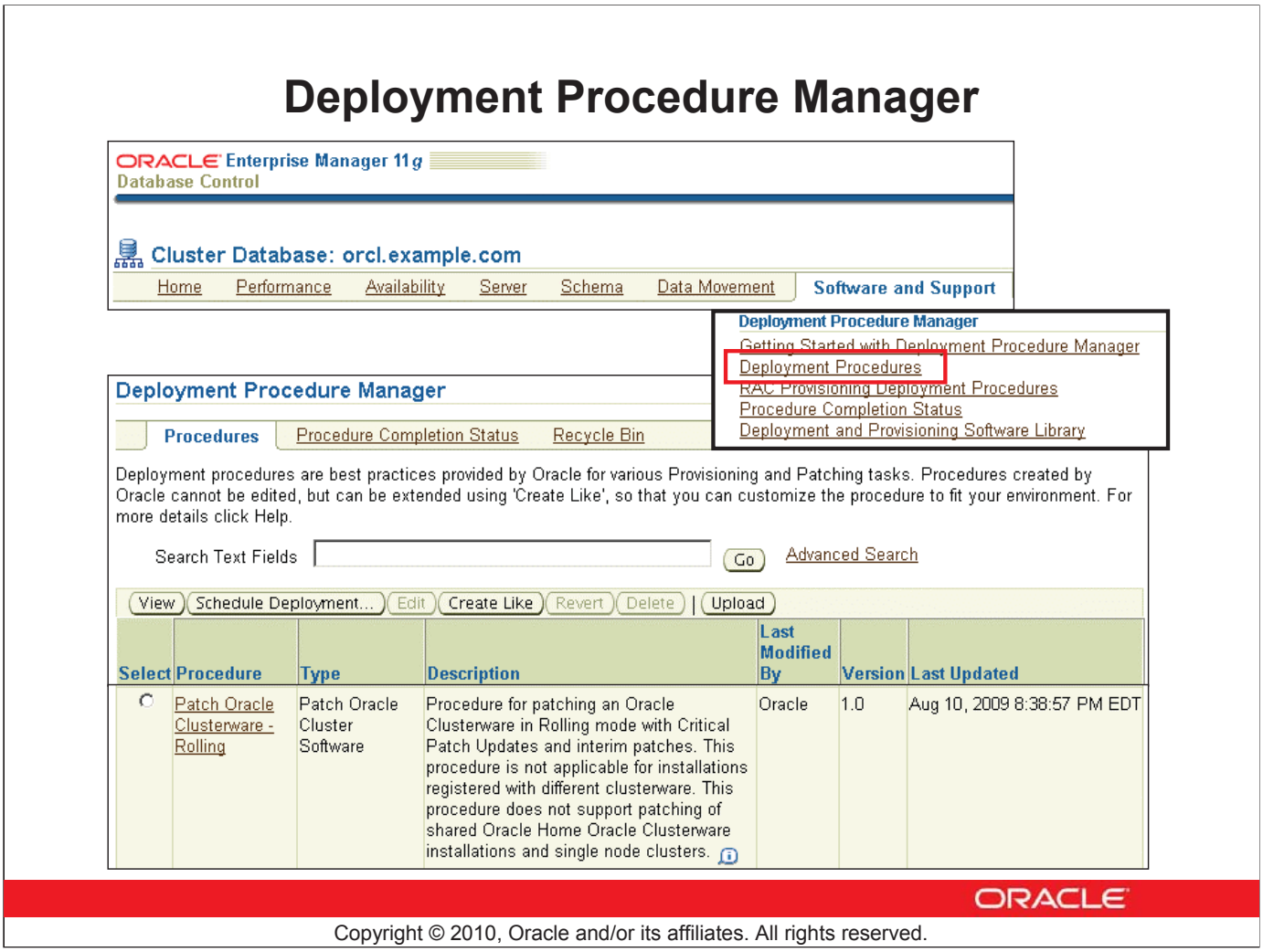
1. On the Database Home page, click Software and Support.
2. On the Software and Support page, select View Patch Cache under the heading Database Software Patching.
3. On the Patch Cache page, click Upload Patch. The “Create Oracle Software Update Component” page appears.
4. Click the Browse button and then navigate to the directory where you saved the patch file to disk. If you are accessing DB Control from a browser running on a client computer, then the patch must be available on that client computer. When you have located the patch file, click the file name to select the file, then click Open. This places the file name in the Patch File field.
5. In the Patch Attributes section of the “Create Oracle Software Update Component” page, enter the following information about the patch you are uploading:
   - Patch number
   - The type of patch (individual or patchset)
   - The date that the patch was created

Uploading Patches (continued)

- The patch description
- The product family for the patch (for example, Oracle Database)
- The product that the patch applies to (for example, RDBMS Server)
- The release version of the product
- The platform that your database runs on
- The language to use (for example, American English)
- (Optional) A description of the patch

You can find most of this information in the patchset's `README` file.

6. When you have finished entering the information, select "Add Patch File to Patch Cache" and click Upload. The Patch Cache page appears again, with a confirmation message.
7. The patch now appears in the Patch Cache.



Deployment Procedure Manager

The “Deployment Procedure Manager” page shows several procedures that can be scheduled to apply patches.

To navigate to the “Deployment Procedure Manager” page, start at the database home page, click the Software and Support tab. On the Software and Support page, click Deployment Procedures. Then you can select a procedure and schedule deployment.

Reduced Down-Time Patching for Cluster Environments

Patching Oracle Clusterware and Oracle RAC can be completed without taking the entire cluster down.

- OPatch now can apply patches in multinode, multipatch fashion.
- OPatch detects whether the database schema is at an earlier patch level than the new patch, and runs SQL commands to bring the schema up to the new patch level.
- OUI installs patchsets as out-of-place upgrades, reducing the down time required for patching.

ORACLE

Reduced Down-Time Patching for Cluster Environments

Patching Oracle Clusterware and Oracle RAC can be completed without taking the entire cluster down. In many cases, patching can be performed with zero down time. This also allows for out-of-place upgrades to the cluster software and Oracle Database, reducing the planned maintenance down time required in an Oracle RAC environment.

OPatch can now apply patches in multinode, multipatch fashion. OPatch will not start up instances that have a nonrolling patch applied to it if other instances of the database do not have that patch. OPatch also detects whether the database schema is at an earlier patch level than the new patch, and it runs SQL commands to bring the schema up to the new patch level.

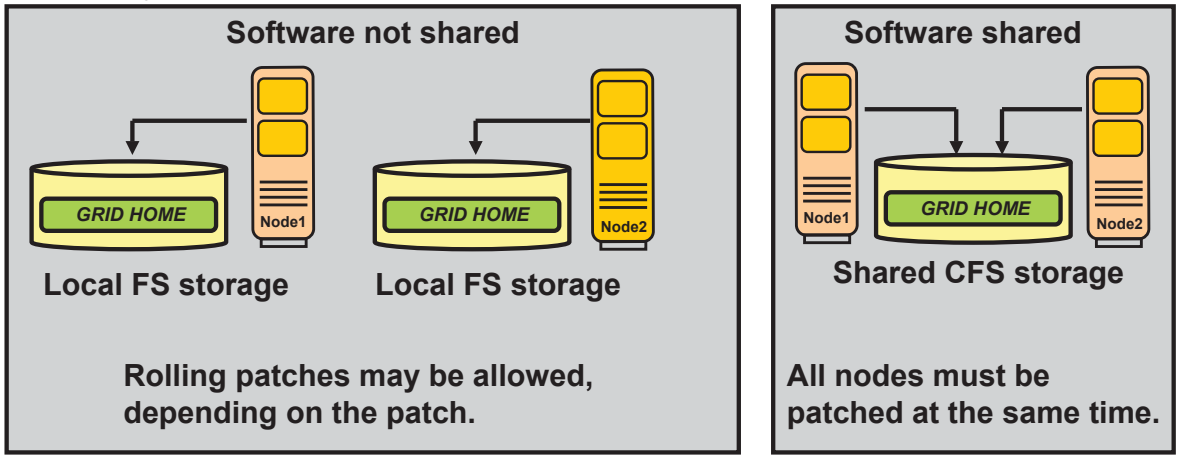
You can use `srvctl` to shut down the Oracle software running within an Oracle home, in preparation for patching. Oracle Grid Infrastructure patching is automated across all nodes, and patches can be applied in a multinode, multipatch fashion.

Patchsets are now installed as out-of-place upgrades to the Grid Infrastructure software (Oracle Clusterware and Automatic Storage Management) and Oracle Database. This reduces the down time required for planned outages for patching.

Rolling Patches

A rolling patch allows one node at a time to be patched, while other nodes continue to provide service. It:

- Requires distinct software homes for each node
- Allows different versions to coexist temporarily
- May not be available for all patches



ORACLE

Rolling Patches

By default, Oracle supports rolling upgrade for patches for Oracle Clusterware. This is available for patchsets, bundle patches, and most individual patches. A rolling patch allows one node to be patched to the latest version, while other nodes continue to use the older version and provide business service.

Rolling patches are enabled by using a locally accessible, nonshared file system to store the software files, known as Grid home. Each node maintains its own copy. With this technique on a 50-node cluster, there would be 50 copies of the Oracle Clusterware software—that is, 50 copies requiring disk space and 50 copies to update. This technique enables rolling patches and rolling upgrades by allowing two versions in the cluster at the same time.

Rolling patches cannot be done when the Oracle Clusterware software files are stored in a shared cluster file system in which a single copy of the software is shared among all nodes. A single copy requires much less disk space and a single copy to patch or upgrade. However, to patch or upgrade, the software must be stopped. Stopping the Oracle Clusterware software also requires all databases, applications, and services that depend on Oracle Clusterware to be stopped. This technique requires a complete outage with down time to patch or upgrade.

Note: A patchset that can be rolled for the clusterware may not be able to be rolled for the RDBMS.

Checking Software Versions

- With rolling patches, the software version may be temporarily newer than the active version.
  - To check the software version on a single node:

```
$ crsctl query crs softwareversion [hostname]
```

  - To check the active version of the cluster:

```
$ crsctl query crs activeversion
```

- Different versions should exist only while applying a patch.

ORACLE

Checking Software Versions

When a rolling patch or upgrade is being performed, two versions of the software will temporarily coexist in the cluster. The software version is the latest version that is installed on an individual node. You can check the software version registered in the OCR with the following command:

```
$ crsctl query crs softwareversion
Oracle Clusterware version on node [host01] is [11.1.0.7.0]
```

The active version is the lowest version anywhere in the cluster. It applies to the cluster and not an individual node. The active version is not updated until the last node has been updated to the newest version. You can check the active version with the following command:

```
$ crsctl query crs activeversion
Oracle Clusterware active version on the cluster is
[11.1.0.6.0]
```

Permanently operating Oracle Clusterware at different versions is not supported. This is allowed only for a short duration—that is, the time it takes to apply the patchset or patch to the cluster.

Note: The version of Oracle Clusterware must be greater than the version running other Oracle products such as the Real Application Clusters (RAC) database and ASM software versions.

Installing a Rolling Patchset with OUI

On one node of the cluster, perform the following steps:

1. Read the latest release notes for the patchset.
2. Shut down all applications that depend on Oracle Clusterware.
3. Stop the Oracle Clusterware node applications, relocating the Virtual IP (VIP) to another node.

```
$ srvctl stop nodeapps -n host01 -r
```

4. Start the newer OUI located from the patchset.

```
$ cd patchset_directory/Disk1
$ ./runInstaller
```

ORACLE

Installing a Rolling Patchset with OUI

After downloading the latest patchset, unzip the distribution into a directory that is accessible by the first node that will be patched. The patchset does not need to be on a shared file system. The installer will distribute the patchset software to the other nodes in the cluster.

Step 1: Read the platform-specific release notes. The patchset is specific to an operating system platform and should contain platform-specific release notes. The outline being presented here is generic, and specific operating systems may contain different steps.

Step 2: It is necessary to stop all applications on the first node that depend on Oracle Clusterware such as database instances, database listeners, ASM instances, and any other applications. The database administrator may choose to disable the connection mechanisms to the first node and have users that are currently connected to the node to log out, log back in, and get redistributed to other nodes before stopping the applications.

Step 3: Stop the Oracle Clusterware node applications with the `srvctl` utility as illustrated in the slide. The `-r` option relocates the Virtual IP (VIP) to other nodes to avoid connection timeouts. If the OCR and voting files are in ASM, use the `crsctl stop crs` command.

Step 4: Always run the latest version of the installer from the unzipped distribution instead of the older version that can be found in the existing clusterware software. OUI is used to install patchsets and must be invoked from a graphical-capable session. The OUI utility must be invoked by the user account that owns the Oracle Clusterware software, not the `root` user.

Patchset OUI

Patchset OUI

Note: The instructions here are general because there were no patchsets available at the time of this writing for Oracle 11*g* Release 2. Check the instructions that come with the patchset.

Invoke OUI from the patchset directory and follow the instructions. In the Welcome window, you may verify a list of all the products that are currently installed on the machine by clicking the Installed Products button. Click the Next button to install the patchset.

The Source window may be displayed, so you can specify the directory where the patchset resides. The Install Location window asks for the Oracle Base directory and the Oracle Home name and directory; the default values depend on the environment variables that are set and the applications currently installed. A product selection window may appear because the patchset contains both RDBMS and clusterware patches.

For an existing cluster, the Specify Hardware Cluster Installation Mode window shows all nodes automatically selected and the option to change this is not allowed. The software will be distributed to each node in the cluster, and no other actions will be taken on those nodes that continue to provide business services.

In the Summary window, all nodes are listed, but software distribution is the only action that is being performed to the other nodes. The remaining nodes can continue to run the Oracle Clusterware stack and applications, providing continuous business service until they are ready to be brought down.

Installing a Rolling Patchset with OUI

As the `root` user, on one node of the cluster, perform the following steps:

5. Shut down the Oracle Clusterware and Oracle daemons.

```
# crsctl stop crs
```

6. Run the `root` upgrade script.

```
# cd /u01/app/11.2.0/grid/install
# ./root*.sh
```

7. Start up all dependent applications that were shut down in step 2.
8. Repeat all steps except step 4 (`runInstaller`) on each node.

ORACLE

Installing a Rolling Patchset with OUI

On the first node that the installation was run on, log in as the `root` user and invoke the `crsctl` utility to stop the clusterware stack. Before stopping the clusterware stack, the software version will still report the old version.

After the clusterware stack has been successfully stopped, invoke the script named in the OUI instructions to perform the update. This script restarts the clusterware stack at the end and the software version then reports the newer version. The active version remains at the older version until all nodes have been updated.

After the clusterware stack has been restarted, all the dependent applications that were shut down on this one node can be restarted. Repeat all the steps on the remaining nodes, except step 4 (`runInstaller`). The software has already been distributed to each node in the cluster. The updates for rest of the nodes need to be completed in a timely manner. Running the nodes of an Oracle Clusterware environment at different software versions for an extended length of time is not supported.

OPatch: Overview

OPatch is a Java-based utility that allows the application and rolling back of interim patches. It:

- Is used for CRS BP and MLR patches
- Supports rolling-patch application for Oracle Clusterware
- Maintains an inventory of the patches that are installed
- Does not require CRS patches to be relinked
- Is invoked as the Grid Infrastructure software owner
- Requires special prescripts and postscripts to be invoked for CRS BP:
  - `prerootpatch.sh` (as `root`)
  - `prepatch.sh` (as the clusterware owner)
  - `postpatch.sh` (as the clusterware owner)
  - `postrootpatch.sh` (as `root`)

ORACLE

OPatch: Overview

OPatch is a Java-based utility that allows the application and rolling back of interim patches to an Oracle product. It is not used to apply patchsets. OPatch is included with the Oracle Clusterware 11*g* installation. When patching an Oracle Clusterware installation using the rolling-patch method, the user is prompted for which nodes to patch during the patch installation process. A rolling patch is identified inside the patch and cannot be enabled when invoking the patch tool. OPatch also supports a minimum down-time mode. An inventory of all patches that have been installed is maintained by OPatch.

Patching Oracle Clusterware has special considerations: Part of the software is owned by `root` and part is owned by the account used to install the product. To patch the `root`-owned components, the clusterware software will need to be unlocked by running special prepatch scripts. After patching, the clusterware software will need to be returned to its previous secure settings by running postpatch scripts. These script can be found in the patch directory. The additional scripts include the following:

- `prerootpatch.sh`:
  - Confirms that the script is invoked as the `root` user
  - Verifies that Oracle Clusterware is stopped on the local node
  - Checks that the correct parameters were specified

OPatch: Overview (continued)

- Unlocks the Oracle Clusterware software, setting the ownership of some files to the non-`root` user, and changes permissions

- `prepatch.sh`:
  - Checks that the parameters are correct
  - Changes the permissions on selected files
  - Searches for variables from the installation included in the existing files
  - Verifies that the values were obtained for the variables
  - Saves the variables and values in the *`Grid_home`*`/install/params.crs` file
- `postpatch.sh`:
  - Verifies that the *`Grid_home`*`/install/params.crs` file has the correct format
  - Parses and replaces the correct values in the files sourcing from `params.crs`
  - Resets the permissions on selected files
- `postrootpatch.sh`:
  - Locks the executables and directories by changing the ownership to `root`
  - Copies the initialization wrapper scripts to the correct location
  - Updates `/etc/inittab` so that the new files are used to start Oracle Clusterware
  - Waits for Oracle Clusterware to start

Note: With the June 2010 PSU, `prerootpatch.sh`, `prepatch.sh`, `postpatch.sh` and `postrootpatch.sh` have been replaced by `rootcrs.pl` although `prepatch.sh` and `postpatch.sh` are still used for database homes.

OPatch: General Usage

- To define the ORACLE_HOME or -oh option on all commands:

```
$ export ORACLE_HOME=/u01/app/11.2.0/grid
$ opatch command [options]
```

or

```
$ opatch command -oh /u01/app/11.2.0/grid [options]
```

- To obtain help with the OPatch syntax:

```
$ opatch command -help
```

- To check whether a patch supports a rolling application (Run from the patch directory.):

```
$ opatch query -all | grep -i Rolling
```

ORACLE

OPatch: General Usage

The OPatch utility requires that the ORACLE_HOME environment variable be defined or that the value of ORACLE_HOME be passed as an argument on the command line with the -oh option. For the case of Oracle Clusterware, ORACLE_HOME refers to the installation directory for Oracle Clusterware, not the location of other Oracle products that may be installed. In general, ORACLE_HOME refers to the home for the product to be patched.

The OPatch documentation can be found in the *Grid_home*/OPatch/docs directory, which includes the *OPatch Users Guide* and the *OPatch Prerequisite Users Guide*. The utility contains help for its syntax by using the -help option as follows:

```
opatch -help
opatch apply -help
opatch lsinventory -help
opatch rollback -help
opatch prereq -help
opatch util -help
```

In general, CRS BPs and CRS MLR patches can be applied in a rolling fashion—that is, one node at a time. However, it is still important to check each patch for exceptions to this rule. To verify that a patch supports rolling applications, unzip the downloaded patch into a directory of your choosing and, from that directory, issue the following command:

```
$ORACLE_HOME/OPatch/opatch query -is_rolling_patch <patch_location>
```

Before Patching with OPatch

- Check the current setting of the `ORACLE_HOME` variable.
- Back up the directory being patched with an OS utility or Oracle Secure backup.
- Stage the patch to each node.
- Update the `PATH` environment variable for the `OPatch` directory.

Before Patching with OPatch

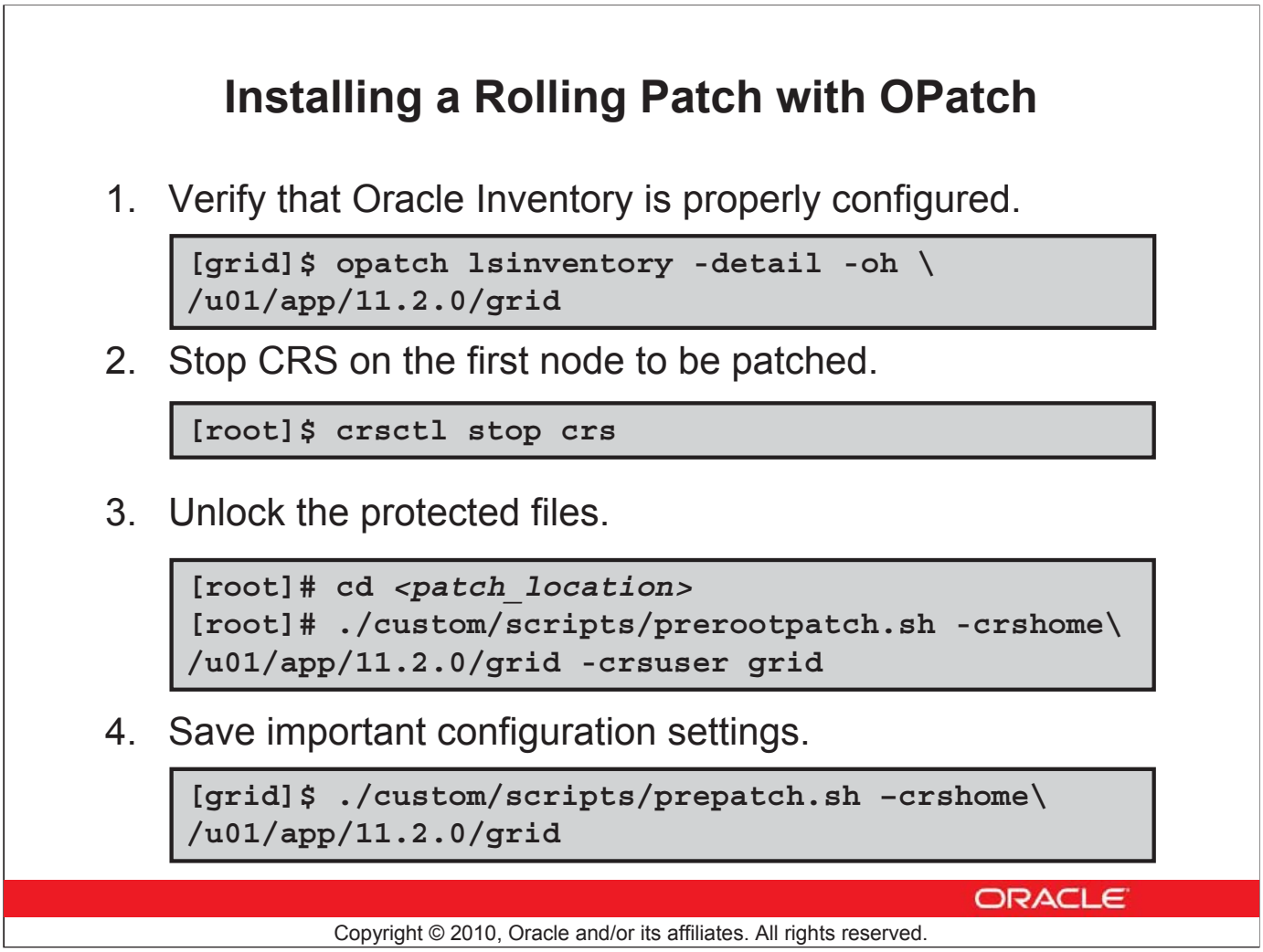
The Oracle Patching utility, OPatch, verifies that the `ORACLE_HOME` environment variable names an actual directory. You should verify that the `ORACLE_HOME` variable is set to the Oracle home of the product you are trying to patch.

It is best practice to back up the software directory that you are patching before performing any patch operation. This applies to Oracle RAC, ASM, or Oracle Clusterware software installation directories. The backup should include the Oracle Inventory directory as well.

If you manually download the patch and use OPatch to install the patch, you must stage the patch on each node. If you use Enterprise Manager to download the patch and you selected all the nodes in your cluster as targets for the patch, then the patch is automatically staged on those nodes.

The `opatch` binary file is located in the `$ORACLE_HOME/OPatch` directory. You can either specify this path when executing OPatch, or you can update the `PATH` environment variable to include the `OPatch` directory. To change the `PATH` variable on Linux, use:

```
$ export PATH=$PATH:$ORACLE_HOME/OPatch
```



Installing a Rolling Patch with OPatch

The OPatch utility can patch all cluster nodes simultaneously with a single invocation of the command. However, this would require Oracle Clusterware to be stopped on all nodes simultaneously. To avoid complete outage, patches can be applied in a rolling fashion to the local node first, and later to each successive node. To apply a rolling patch with OPatch, start in step 1 by verifying that Oracle Inventory can be located and properly configured with the following command:

```
opatch lsinventory -detail -oh Grid_home
```

If the `ORACLE_HOME` environment variable has been defined, it is not necessary to include the `-oh` option. For step 2, stop the Oracle Clusterware process stack on the local node with the following command:

```
crsctl stop crs
```

For step 3, log in to the `root` user account and unlock the protected files with the following command:

```
prerootpatch.sh -crshome Grid_home -crsuser <Grid
owner>
```

For step 4, save important configuration settings to prevent them from being overwritten by the patch using the following command:

```
prepatch.sh -crshome Grid_home
```

The *Grid_home*`/install/params.crs` file will have been created.

Installing a Rolling Patch with OPatch

5. Patch the CRS installation on the first node only.

```
[grid]$ opatch apply -local -oh Grid_home\
 patch_location
```

6. Apply configuration settings to the patched files.

```
[grid]$ ./custom/scripts/postpatch.sh -crshome\
Grid_home
```

7. Restore the lock to the protected files.

```
[root]# ./custom/scripts/postrootpatch.sh\
 -crshome Grid_home
```

8. Verify patch installation.

```
[grid]$ opatch lsinventory -detail -oh Grid_home
```

9. Repeat steps 1–8 on each node, one at a time.

ORACLE

Installing a Rolling Patch with OPatch (continued)

For step 5, patch only the local Grid installation on the first node with the following command:

```
opatch apply –local –oh grid_home patch_location
```

The patch location is the name of the directory when the patch was unzipped. If a path name is not given, the current directory is used. For step 6, apply the configuration settings that were saved in step 4 to the files that have been overwritten by the patch with the following command:

```
postpatch.sh –crshome <grid_home>
```

For step 7, it is necessary to restore the lock of the Oracle Clusterware software by setting the ownership of selected files back to the `root` user and setting the permissions accordingly by using the following command as the `root` user:

```
postrootpatch.sh –crshome Grid_home
```

For step 8, verify the patch installation with the following command:

```
opatch lsinventory –detail –oh Grid_home
```

The eight steps can then be repeated on each node of the cluster, one node at a time, to roll the patch throughout the cluster.

Note: A patch may contain both Grid and RDBMS patch elements. Refer to the `README.txt` file in the patch for additional instructions.

Installing a Patch with Minimum Down Time with OPatch

Actions:

- The local node is always patched first.
- The local node is used as a base to patch the other nodes.
- The user is prompted for the first set of nodes to patch.
- For each node in the first set:
  - Stop the instance.
  - The patch is propagated.
  - Patch the node.
- Shut down Clusterware on the remaining nodes.
- The patch is propagated to the remaining nodes.
- Inventory is updated.
- Start up the first set before patching each node of the second set.

ORACLE

Installing a Patch with Minimum Down Time with OPatch

In minimum down-time patching, the nodes are divided into two sets. One set of nodes is shut down and the patch is applied to those nodes. After the first set of nodes has been patched, the second set of nodes is shut down. The first set of nodes is then restarted and the patch is applied to the second set of nodes. After the patch has been applied to the second set of nodes, those nodes are restarted. This method leads to less down time for Oracle RAC, compared to having all the nodes shut down at the same time.

When you use the minimum down-time patching method, the following actions occur:

- The local node is always patched first.
- The local node is used as a base to patch the other nodes.
- The user is prompted for the first set of nodes to patch.
- For each node in this first set, the user is asked to stop the instance and then the patch is propagated and applied to that node before continuing to the next node.
- When the first set of nodes has been patched, the user is asked to shut down Clusterware on the remaining nodes.
- The instances are stopped on the last set of remote nodes.
- The patch is propagated to the last set of nodes and the inventory is updated.
- You can then start up the patched nodes (the first set of nodes) before patching the remaining nodes.

Quiz

Which tools can be used to install a patchset?

1. Oracle Universal Installer
2. OPatch
3. Enterprise Manager Database Console
4. Database Configuration Assistant

Answer: 1,3

In Oracle 11*g* Release 2, the Oracle Universal Installer or Enterprise Manager Database Console can be used.

Quiz

Patching an Oracle Clusterware environment has special consideration due to file ownerships and permissions.

1. True
2. False

ORACLE

Answer: 1

The Oracle Home directory for Oracle Clusterware and many of the files in the directory are owned by `root`. The permissions and ownership have to be changed before patching and returned to the original state after patching.

Summary

In this module, you should have learned how to:

- Describe the types of patches available
- Design for rolling patches
- Compare software versions with the active version
- Install a patchset with the Oracle Universal Installer (OUI) utility
- Install a patch with the `opatch` utility

ORACLE

6

Troubleshooting Oracle Clusterware

Objectives

After completing this lesson, you should be able to:

- Locate Oracle Clusterware log files
- Gather all log files using `diagcollection.pl`
- Enable resource debugging
- Enable component-level debugging
- Enable tracing for Java-based tools
- Troubleshoot the Oracle Cluster Registry (OCR) file

ORACLE

Golden Rule in Debugging Oracle Clusterware

- Always make sure that your nodes have exactly the same system time to:
  - Facilitate log information analysis
  - Ensure accurate results when reading GV$ views for Oracle Real Application Clusters (RAC) database instances
  - Avoid untimely instance evictions
- The best recommendation is to synchronize nodes using Network Time Protocol (NTP).
  - Modify the NTP initialization file to set the -x flag, which prevents time from being adjusted backward.

```
# vi /etc/sysconfig/ntp
OPTIONS="-x -u ntp:ntp -p /var/run/ntpd.pid"
```

- If NTP is not used, Clusterware will automatically configure Cluster Time Synchronization Service (CTSS).

ORACLE

Golden Rule in Debugging Oracle Clusterware

It is recommended that you set up Network Time Protocol (NTP) on all cluster nodes before you install Oracle Clusterware. This synchronizes the clocks among all nodes and facilitates the analysis of tracing information based on time stamps as well as results from queries issued on GV$ views when using an Oracle RAC database instance. These views are used by the database administrator to view consolidated database information from each node in the cluster. For Oracle Enterprise Linux, NTP is configured using the /etc/ntp.conf file. Edit the file and add the following entries:

```
server  name01.example.com   #Server with atomic clock
server  name02.example.com   #Server with less accuracy
restrict name01.example.com mask 255.255.255.255 nomodify notrap noquery
restrict name02.example.com mask 255.255.255.255 nomodify notrap noquery
```

You can start the NTP service with the service ntpd start command. Enable NTP to start at each boot with the chkconfig ntpd on command. The ntpq utility can be used to check the performance of the NTP servers.

If you are using NTP, and you prefer to use it instead of CTSS, then you need to modify the NTP initialization file to set the -x flag, which prevents time from being adjusted backward. Restart the network time protocol daemon after you complete this task.

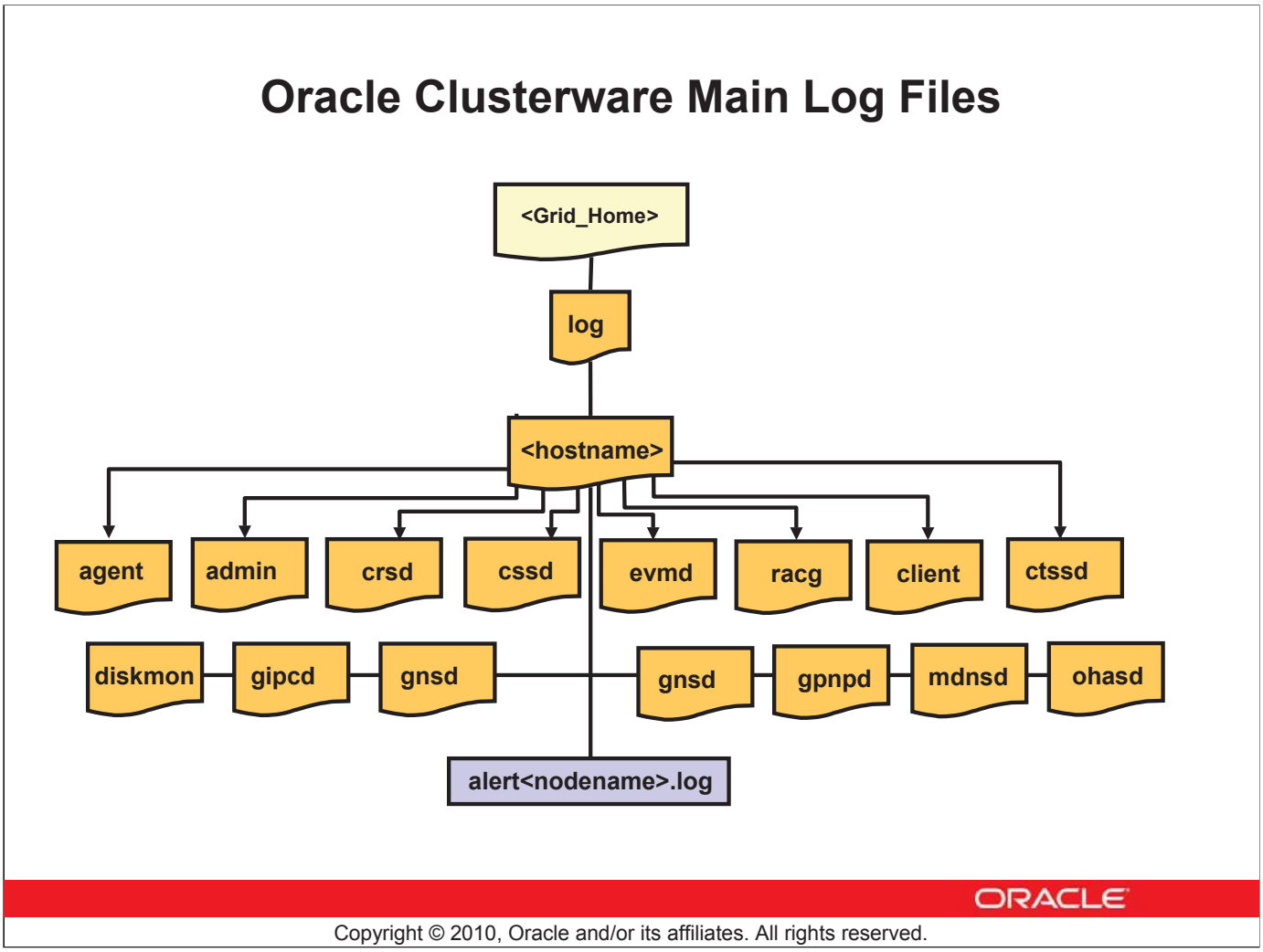
Golden Rule in Debugging Oracle Clusterware (continued)

To do this, on Oracle Enterprise Linux, Red Hat Linux, and Asianux systems, edit the `/etc/sysconfig/ntpd` file to add the `-x` flag, as in the following example:

```
OPTIONS="-x -u ntp:ntp -p /var/run/ntpd.pid"
```

If NTP is not used, Clusterware will automatically configure Cluster Time Synchronization Service (CTSS) and start the `octcssd.bin` daemon on the cluster nodes.

Note: Adjusting clocks by more than five minutes can cause instance evictions. It is strongly advised to shut down all instances before date/time adjustments.



Oracle Clusterware Main Log Files

Oracle Clusterware uses a unified log directory structure to consolidate the Oracle Clusterware component log files. This consolidated structure simplifies diagnostic information collection and assists during data retrieval and problem analysis.

The slide shows you the main directories used by Oracle Clusterware to store its log files:

- Cluster Ready Service (CRS) logs are in `<Grid_Home>/log/<hostname>/crsd/`. The `crsd.log` file is archived every 10 MB (`crsd.l01`, `crsd.l02`, …).
- Cluster Synchronization Service (CSS) logs are in `<Grid_Home>/log/<hostname>/cssd/`. The `cssd.log` file is archived every 20 MB (`cssd.l01`, `cssd.l02`, …).
- Event Manager (EVM) logs are in `<Grid_Home>/log/<hostname>/evmd`.
- SRVM (`srvctl`) and OCR (`ocrdump`, `ocrconfig`, `ocrcheck`) logs are in `<Grid_Home>/log/<hostname>/client/` and `$ORACLE_HOME/log/<hostname>/client/`.
- Important Oracle Clusterware alerts can be found in `alert<nodename>.log` in the `<Grid_Home>/log/<hostname>` directory.
- Oracle Cluster Registry tools (`ocrdump`, `ocrcheck`, `ocrconfig`) logs can be found in `<Grid_Home>/log/<hostname>/client`.

In addition, important Automatic Storage Management (ASM)–related trace and alert information can be found in the `<Grid_Base>/diag/asm/+asm/+ASMn` directory, specifically the log and trace directories.

Diagnostics Collection Script

- The `<Grid_Home>/bin/diagcollection.pl` script is used to collect important log files and must be run as `root`.
- It generates the following files in the local directory:
  - `crsData_<hostname_date_time>.tar.gz`
  - `ocrData_<hostname_date_time>.tar.gz`
  - `coreData_<hostname_date_time>.tar.gz`
  - `osData_<hostname_date_time>.tar.gz`

```
# /u01/app/11.2.0/grid/bin/diagcollection.pl --collect
Production Copyright 2004, 2008, Oracle.  All rights reserved
Cluster Ready Services (CRS) diagnostic collection tool
The following diagnostic archives will be created in the local directory.
crsData_host01_20090729_1013.tar.gz -> logs,traces and cores from CRS home.
Note: core files will be packaged only with the --core option.
ocrData_host01_20090729_1013.tar.gz -> ocrdump, ocrcheck etc
coreData_host01_20090729_1013.tar.gz -> contents of CRS core files
osData_host01_20090729_1013.tar.gz -> logs from Operating System
...
```

ORACLE

Diagnostics Collection Script

Use the `diagcollection.pl` script to collect diagnostic information from an Oracle Grid Infrastructure installation. The diagnostics provide additional information so that Oracle Support can resolve problems. This script is located in `<Grid_Home>/bin`. Before executing the script, you must be logged in as `root`. The example in the slide shows you how to invoke the script to collect diagnostic information. When invoked with the `--collect` option, the script generates, in the local directory, the four files mentioned in the slide.

- `coreData...tar.gz` contains core files and core file analysis extracted to text files.
- `crsData...tar.gz` contains log files from `<Grid_Home>/log/<hostname>`.
- `ocrData...tar.gz` files contain the results of an `ocrdump` and `ocrcheck`, and the list of OCR backups.
- `osData...tar.gz` contains `/var/log/messages` and the associated archived files.

You can also invoke the script with the `--clean` option to clean out the files generated from a previous run in your local directory. Alternatively, you can invoke the script to capture just a subset of the log files. You can do so by adding extra options after the `--collect` option: `--crs` for collecting Oracle Clusterware logs, `--core` for collecting core files, or `--all` for collecting all logs. The `--all` option is the default.

Cluster Verify: Overview

- Use the Cluster Verification Utility (CVU) to verify that you have a well-formed cluster for Oracle Grid Infrastructure and RAC:
  - Installation
  - Configuration
  - Operation
- You can perform a full stack verification.
- It uses a nonintrusive verification.
- Diagnostic mode seeks to establish a reason for the failure of any verification task.
- You can generate fixup scripts with some CVU commands by using the `-fixup` flag.

ORACLE

Cluster Verify: Overview

Cluster Verification Utility (CVU) is provided with Oracle Clusterware and Oracle Database 10*g* Release 2 (10.2) and later with the RAC option. The purpose of CVU is to enable you to verify during setup and configuration that all components required for a successful installation of Oracle Grid Infrastructure or Oracle Grid Infrastructure and a RAC database are installed and configured correctly, and to provide you with ongoing assistance any time that you need to make changes to your cluster or RAC database.

There are two types of CVU commands:

- Stage commands are CVU commands used to test system setup and readiness for successful software installation, database creation, or configuration change steps. These commands are also used to validate successful completion of specific configuration steps.
- Component commands are CVU commands used to check individual cluster components and determine their state.

In addition, you can use CVU to verify a particular component while the stack is running or to isolate a cluster subsystem for diagnosis. During the diagnostic mode of operation, CVU tries to establish a reason for the failure of any verification task to help diagnose a problem.

CVU can generate fixup scripts that perform system configuration needed for a successful installation in addition to identifying system issues that can cause installation failures. CVU provides additional checks to address installation, configuration, and operational issues.

Cluster Verify Components

- An individual subsystem or a module of the RAC cluster is known as a component in CVU.
- The availability and integrity of a cluster component can be verified.
- Various components—some simple such as a specific storage device, others complex such as the Oracle Clusterware stack—include:
  - Space availability
  - Shared storage accessibility
  - Node connectivity
  - Cluster File System integrity
  - Oracle Clusterware integrity
  - Cluster integrity
  - Administrative privileges
  - Peer compatibility
  - System requirements

```
$ cluvfy comp -list
```

ORACLE

Cluster Verify Components

CVU supports the notion of component verification. The verifications in this category are not associated with any specific stage. A component can range from basic, such as free disk space, to complex (spanning over multiple subcomponents), such as the Oracle Clusterware stack. Availability, integrity, or any other specific behavior of a cluster component can be verified. You can list verifiable CVU components with the `cluvfy comp -list` command:

nodereach	Checks node reachability	**peer**	Compares properties with peers
nodecon	Checks node connectivity	**ha**	Checks HA integrity
cfs	Checks CFS integrity	**asm**	Checks ASM integrity
ssa	Checks shared storage	**acfs**	Checks ACFS integrity
space	Checks space availability	**olr**	Checks OLR integrity
sys	Checks minimum requirements	**gpnp**	Checks GPnP integrity
clu	Checks cluster integrity	**gns**	Checks GNS integrity
clumgr	Checks cluster manager integrity	**scan**	Checks SCAN configuration
ocr	Checks OCR integrity	**ohasd**	Checks OHASD integrity
admprv	Checks administrative privileges	**crs**	Checks CRS integrity
software	Checks software distribution	**vdisk**	Checks Voting Disk Udev settings
clocksync	Checks clock synchronization		
nodeapp	Checks node applications' existence		

Cluster Verify Locations

- Download it from OTN:
  - Create a local directory.
  - Copy and extract `cvu_<OS>.zip`.
  - Use the most recent version.
- Oracle software DVD:
  - `Disk1` directory
  - `runcluvfy.sh`
- Grid Infrastructure home:
  - `<Grid_Home>/bin/cluvfy`
- Oracle home:
  - `$ORACLE_HOME/bin/cluvfy`

ORACLE

Cluster Verify Locations

Cluster Verification Utility (CVU) was first released in Oracle Clusterware Release 10.2.0.1.0. CVU supports 11*g*R2, 11*g*R1, 10*g*R2, as well as 10*g*R1 for Oracle Clusterware and RAC products. CVU is available in three different forms:

- Available on Oracle Technology Network (OTN) at: http://www.oracle.com/technology/products/database/clustering/cvu/cvu_download_homepage.html
  From there, you need to download the package and unzip it to a local directory (`<cvhome>`). You can use the `cluvfy` command from `<cvhome>/bin`. Optionally, you can set the `CV_DESTLOC` environment variable. This should point to a writable area on all nodes. CVU attempts to copy the necessary bits as required to this location. If this variable is not set, CVU uses `/tmp` as the default.
- Available in 11.2 Oracle software DVD as a packaged version. Make use of `runcluvfy.sh`, which is needed when nothing is installed. You can find it in `Disk1`.
- Installed in both 11.2 Oracle Clusterware and RAC homes. Make use of `cluvfy` if the CRS software stack is installed. If the CRS software is installed, you can find `cluvfy` under `<Grid_Home>/bin`.

Note: For manual installation, you need to install CVU on only one node. CVU deploys itself on remote nodes during executions that require access to remote nodes.

Cluster Verify Configuration File

```
$ cat cvu_config
# Configuration file for Cluster Verification Utility(CVU)
# Version: 011405
#
#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

# Whether X-Windows check should be performed for SSH user equivalence
#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
```


Cluster Verify Configuration File

You can use the CVU's configuration file to define specific inputs for the execution of CVU. The path for the configuration file is `$CV_HOME/cv/admin/cvu_config`. The following is the list of keys supported in `cvu_config`:

- **`CV_ORACLE_RELEASE:`** This is the version of the software to check against.
- **`CV_NODE_ALL:`** If set, this specifies the list of nodes that should be picked up when Oracle Clusterware is not installed and the `-n all` option has been used in the command line.
- **`CV_RAW_CHECK_ENABLED:`** If set to `TRUE`, this enables the check for accessibility of shared SCSI disks on Red Hat Release 3.0 and later. This shared disk accessibility check requires that you install a `cvuqdisk rpm` on all the nodes. By default, this key is set to `TRUE` and shared disk check is enabled.
- **`CV_ASSUME_DISTID:`** This specifies the distribution ID that CVU uses. For example, to make CVU working with SuSE 9 ES, set it to `Pensacola`.
- **`CV_XCHK_FOR_SSH_ENABLED:`** If set to `TRUE`, this enables the X-Windows check for verifying user equivalence with `ssh`. By default, this entry is commented out and X-Windows check is disabled.
- **`CV_TRACELOC:`** To choose the location in which CVU generates the trace files, set this environment variable to the absolute path of the desired trace directory.

Cluster Verify Configuration File (continued)

- **ORACLE_SRVM_REMOTESHELL:** If set, this specifies the location for the `ssh`/`rsh` command to override CVU's default value. By default, this entry is commented out and the tool uses `/usr/sbin/ssh` and `/usr/sbin/rsh`.

Note: If CVU does not find a key entry defined in the configuration file, it searches for the environment variable that matches the name of the key; otherwise, it uses a default.

- **ORACLE_SRVM_REMOTECOPY:** If set, this specifies the location for the `scp` or `rcp` command to override the CVU default value. By default, this entry is commented out and CVU uses `/usr/bin/scp` and `/usr/sbin/rcp`.

If CVU does not find a key entry defined in the configuration file, it searches for the environment variable that matches the name of the key. If the environment variable is set, CVU uses its value. Otherwise, it uses a default value for that entity.

To provide CVU with a list of all the nodes of a cluster, you can use the `-n all` option while executing a command.

Cluster Verify Output: Example

```
$ cluvfy comp crs -n all -verbose

Verifying CRS integrity

Checking CRS integrity...
The Oracle clusterware is healthy on node "host03"
The Oracle clusterware is healthy on node "host02"
The Oracle clusterware is healthy on node "host01"

CRS integrity check passed

Verification of CRS integrity was successful.
```

ORACLE

Cluster Verify Output: Example

The slide shows you the output of the `cluvfy comp crs -n all -verbose` command. This command checks the complete Oracle Clusterware stack.

Enabling Resource Debugging

- Change the `USR_ORA_DEBUG` resource attribute to `1` for specific resources:

```
# crsctl set log res "ora.host01.vip:1"
```

- After you capture all trace information, change the debug attribute back to 0:

```
# crsctl set log res "ora.host01.vip:0"
```

- You can use an initialization file to configure debugging.
- The initialization file name includes the name of the process that you are debugging (*process_name*`.ini`).
- The file is located in the `<Grid_Home>/log/host_name/admin/` directory.

ORACLE

Enabling Resource Debugging

Oracle Support may request that you enable tracing to capture additional information for problem resolution with Oracle Clusterware resources. Because the procedures described here may affect performance, perform these activities only with the assistance of Oracle Support.

You can enable debugging for Oracle Clusterware resources by issuing `crsctl set log` and `crsctl set trace` commands, using the following syntax:

```
crsctl set {log | trace} resource "resource_name=debugging_level"
```

Run the crsctl set command as the root user, and supply the following information:

- `resource_name`: The name of the resource to debug.
- `debugging_level`: A number from 1 to 5 to indicate the level of detail you want the debug command to return, where 1 is the least amount of debugging output and 5 provides the most detailed debugging output.

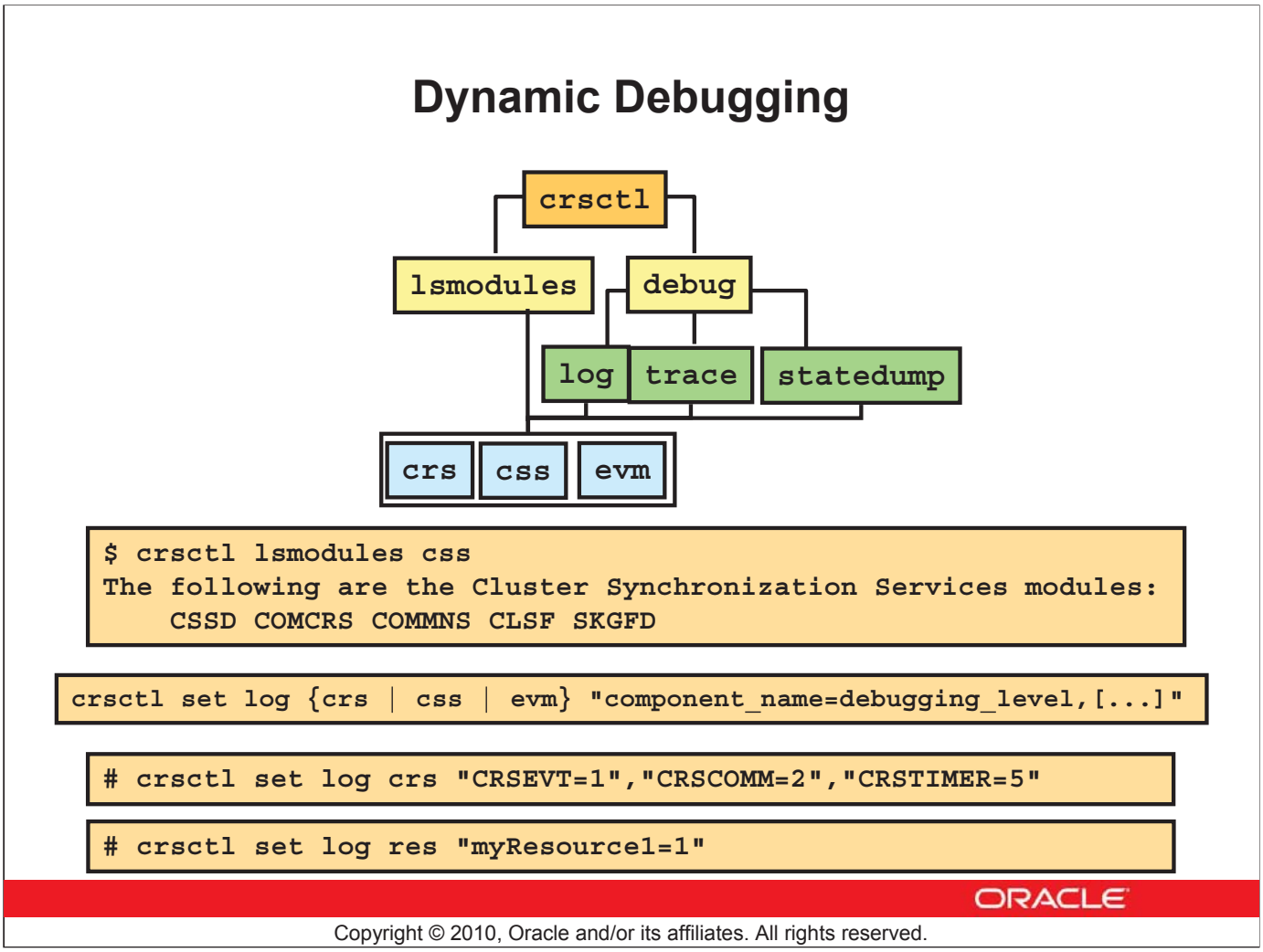
You can dynamically change the debugging level in the crsctl command or you can configure an initialization file for changing the debugging level

The example in the slide enables debugging for the `ora.host01.vip` resource. After you capture all trace information, do not forget to execute the corresponding `crsctl set log res "<resource name>:0"` commands.

Enabling Resource Debugging (Continued)

Another method of enabling resource debugging is the use of an initialization file. This debugging information is stored for use during the next startup. For each process to debug, you can create an initialization file that contains the debugging level.

The initialization file name includes the name of the process that you are debugging (*process_name*.ini). The file is located in the <Grid_Home>/log/host_name/admin/ directory. For example, the name for the CLSCFG debugging initialization file on node1 would be: <Grid_Home>/log/node1/admin/clscfg.ini



Dynamic Debugging

You can use `crsctl set log` commands as the `root` user to enable dynamic debugging for Cluster Ready Services (CRS), Cluster Synchronization Services (CSS), and the Event Manager (EVM), and Clusterware subcomponents. Debugging information remains in the Oracle Cluster Registry (OCR) for use during the next startup. You can also enable debugging for resources. The `crsctl lsmodules crs|css|evm` commands are used to list the module's components that can be used for debugging. The example in the slide lists the ones for `CSS`. Nondefault logging levels are best interpreted with access to the source code and are designed for Oracle Support needs. When asked by Oracle Support, you can use commands such as the following to enable additional logging:

- `crsctl set log <module name> "<component>=<debugging level>"`, where `<module name>` is the name of the module, `CRS`, `EVM`, or `CSS`; `<component name>` is the name of the corresponding component obtained using the `crsctl lsmodules` command, and `<debugging level>` is a level from `0` to `5`
- `crsctl debug statedump crs|css|evm`, which dumps state information for `crs`, `css`, or `evm` modules
- `crsctl set log res "<res_name>=<debugging_level>"`, which sets logging for the specified resource

The example in the slide shows you how to dynamically enable additional logging (level `5`) for the following CRS components: `CRSEVT`, `CRSAPP`, `CRSTIMER`, and `CRSRES`.

Dynamic Debugging

Oracle Clusterware (CRS) Components

Component Name	Description
`CRSUI`	User interface module
`CRSCOMM`	Communication module
`CRSRTI`	Resource management module
`CRSMAIN`	Main module/driver
`CRSPLACE`	CRS placement module
`CRSAPP`	CRS application
`CRSRES`	CRS resources
`CRSOCR`	Oracle Cluster Registry interface
`CRSTIMER`	Various timers related to CRS
`CRSEVT`	CRS EVM/event interface module
`CRSD`	CRS daemon

ORACLE

Dynamic Debugging (Continued)

The table above describes the functions performed by each CRS component.

Enabling Tracing for Java-Based Tools

To enable tracing for `cluvfy`, `netca`, and `srvctl`, set `SRVM_TRACE` to `TRUE`:

```
$ export SRVM_TRACE=TRUE
$ srvctl config database -d orcl > /tmp/srvctl.trc
$ cat /tmp/srvctl.trc
...
[main] [ 2009-09-16 00:58:53.197 EDT ]
[CRSNativeResult.addRIAttr:139]  addRIAttr: name 'ora.orcl.db 3
1', 'USR_ORA_INST_NAME@SERVERNAME(host01)':'orcl1'
[main] [ 2009-09-16 00:58:53.197 EDT ]
[CRSNativeResult.addRIAttr:139]  addRIAttr: name 'ora.orcl.db 3
1', 'USR_ORA_INST_NAME@SERVERNAME(host02)':'orcl2'
[main] [ 2009-09-16 00:58:53.198 EDT ]
[CRSNativeResult.addRIAttr:139]  addRIAttr: name 'ora.orcl.db 3
1', 'USR_ORA_INST_NAME@SERVERNAME(host03)':'orcl3'
[main] [ 2009-09-16 00:58:53.198 EDT ]
[CRSNative.searchEntities:857]  found 3 ntitie
...
```

ORACLE

Enabling Tracing for Java-Based Tools

All Java-based tools and utilities that are available in RAC are invoked by executing scripts of the same name as the tool or utility. This includes the Cluster Verification Utility (`cluvfy`), the Database Configuration Assistant (`dbca`), the Database Upgrade Assistant (`dbua`), the Net Configuration Assistant (`netca`), and Server Control (`srvctl`). For example, to run the Database Configuration Assistant, enter the `dbca` command.

By default, Oracle enables traces for `dbca` and `dbua`. The resulting log files are written to `$ORACLE_HOME/cfgtoollogs/dbca/` and `$ORACLE_HOME/cfgtoollogs/dbua`, respectively. For `cluvfy` and `srvctl`, you can set the `SRVM_TRACE` environment variable to `TRUE` to make the system generate traces. Traces are written to either log files or standard output. For example, the system writes traces to log files in `<Grid_Home>/cv/log/` for `cluvfy`. However, as shown in the slide, it writes traces directly to the standard output for `srvctl`.

To disable tracing for Java-based tools, unset the `SRVM_TRACE` variable:

```
export SRVM_TRACE=
```

Preserving Log Files Before Wrapping

- Write a shell script to copy log files before they wrap.

```
# Script to archive log files before wrapping occurs
# Written for CSS logs. Modify for other log file types.
CSSLOGDIR=/u01/app/11.2.0/grid/log/host01/cssd
while [ 1 -ne 0 ]; do
   CSSFILE=/tmp/css_`date +%m%d%y"_"%H%M`.tar
   tar -cf $CSSFILE $CSSLOGDIR/*
   sleep 300
done
exit
```

- Execute the script created in the background.

```
# chmod 755 archscript.sh; nohup ./archscript.sh &
```

ORACLE

Preserving Log Files Before Wrapping

Depending on the level of debugging that has been enabled, many of the log files are capable of wrapping, which causes only a limited amount of time-stamp information to be available for debugging. To prevent the loss of information, create a script as illustrated in the slide that will copy all the log files before wrapping can occur. Execute the script in the background to capture the logs while debugging is being used. When it is no longer necessary to capture log files, kill the archiving script that is running.

This technique can be applied to any log file for which it is needed. The logs for CSS are shown only as an example. Oracle Clusterware logs already have a log rotation mechanism to save logs, but with excessive debug levels turned on, they could still wrap and lose information.

Process Roles for Node Reboots

The following processes can evict nodes from the cluster or cause a node reboot:

- `oclskd`: Is used by CSS to reboot a node based on requests from other nodes in the cluster
- `cssdagent` and `cssmonitor`: Monitor node hands and vendor clusterware
- `ocssd`: Monitors the internode's health status
- `hangcheck-timer`: Monitors for machine hangs and pauses
  - Not required for Oracle Clusterware 11g Release 2

ORACLE

Process Roles for Node Reboots

Oracle Clusterware is designed to perform a node eviction by removing one or more nodes from the cluster if some critical problem is detected. A critical problem could be a node not responding via a network heartbeat, a node not responding via a disk heartbeat, a hung or severely degraded machine, or a hung ocssd.bin process. The purpose of this node eviction is to maintain the overall health of the cluster by removing suspect members.

CSS uses the Oracle Clusterware Kill Daemon (`oclskd`) to stop processes associated with CSS group members for which stop requests have come.

The `cssdagent` process is spawned by `ohasd` and is responsible for spawning the `ocssd` process, monitoring for node hangs, and monitoring the `ocssd` process for hangs, and monitoring vendor clusterware (via vmon functionality). This is a multi-threaded process that runs at an elevated priority and runs as the root user.

The `ocssd` process is spawned by the `cssdagent` process. It runs in both vendor clusterware and non-vendor clusterware environments. OCSSD's primary job is inter-node health monitoring and RDBMS instance endpoint discovery. The health monitoring includes a network heartbeat and a disk heartbeat (to the voting files). The `ocssd` process can also evict a node after escalation of a member kill from a client (such as a database LMON process). This is a multi-threaded process that runs at an elevated priority and runs as the Oracle user.

Process Roles for Node Reboots (Continued)

The `cssdmonitor` daemon monitors the `ocssd` daemon for hangs or scheduling issues and can reboot a node if there is a perceived hang. If the `ocssd` daemon is lost, the node will be rebooted.

The hangcheck-timer module is a kernel module on Linux systems that monitors for hangs and pauses where the system resumes after some time, but the clock has not noticed the system's lost time. If a threshold of lost time is exceeded, the module will reboot the node.

Note: While the hangcheck-timer module is still required for Oracle Database 11g Release 1 RAC databases, it is no longer needed for Oracle Database 11g Release 2 RAC.

Determining Which Process Caused Reboot

Log File Locations for Processes Causing Reboots.

- **oclskd**
  - <Grid_Home>/log/<*hostname*>/client/oclskd.log
- **ocssd**
  - /var/log/messages
  - <Grid_Home>/log/<*hostname*>/cssd/ocssd.log
- **cssdagent**
  - <Grid_Home>/log/<*hostname*>/agent/ohasd/oracssdagent_root
- **cssdmonitor**
  - <Grid_Home>/log/<*hostname*>/agent/ohasd/oracssdmonitor_root
- **hangcheck-timer**
  - /var/log/messages

ORACLE

Determining Which Process Caused Reboot

First, determine the time of the node reboot by using the `uptime` command and subtracting the up time from the current system time. The reboot time will be used when examining log files.

When the `OCSSD` daemon is responsible for rebooting a node, a message similar to "Oracle CSSD failure. Rebooting for cluster integrity" is written into the system messages log at `/var/log/messages`. The `cssd` daemon log file that is located at `<Grid_Home>/log/<hostname>/cssd/ocssd.log` may also contain messages similar to `"Begin Dump"` or `"End Dump"` just before the reboot.

If hangcheck-timer is being used, it will provide message logging to the system messages log when a node restart is initiated by the module. To verify whether this process was responsible for the node reboot, examine the `/var/log/messages` file and look for an error message similar to: `"Hangcheck: hangcheck is restarting the machine."`

Other useful log files include the Clusterware alert log in `<Grid_home>/log/<hostname>` and the lastgasp log in `/etc/oracle/lastgasp` or `/var/opt/oracle/lastgasp`.

If no indication of which process caused the reboot can be determined from these files, additional debugging and tracing may need to be enabled.

Note: The `oclsomon` and the `oprocd` background processes have been eliminated in Oracle Database 11*g* Release 2.

Using `ocrdump` to View Logical Contents of the OCR

- To dump the OCR contents into a text file for reading:

```
[grid]$ ocrdump filename_with_limited_results.txt
```

```
[root]# ocrdump filename_with_full_results.txt
```

- To dump the OCR contents for a specific key:

```
# ocrdump -keyname SYSTEM.language
```

- To dump the OCR contents to `stdout` in XML format:

```
# ocrdump -stdout -xml
```

- To dump the contents of an OCR backup file:

```
# ocrdump -backupfile week.ocr
```

ORACLE

Using `ocrdump` to View Logical Contents of the OCR

The `ocrdump` utility can be used to view the OCR content for troubleshooting. The `ocrdump` utility enables you to view logical information by writing the contents to a file or displaying the contents to `stdout` in a readable format. If the `ocrdump` command is issued without any options, the default file name of `OCRDUMPFILE` will be written to the current directory, provided that the directory is writable. The information contained within the OCR is organized by keys that are associated with privileges. Therefore, the `root` user will not see the same results as the clusterware owner. Consider the following:

As `root`, `ocrdump -stdout | wc -l` results in 3355 lines on a test system, and as `grid`, `ocrdump -stdout | wc -l` results in 521 lines on the same system.

The number of lines of information in the OCR depends on many factors such as number of nodes in the cluster and number of resources registered in the OCR. Your numbers are expected to be different.

To determine all the changes that have occurred in the OCR over the previous week, locate the automatic backup from the previous week and compare it to a dump of the current OCR as follows:

```
# ocrdump
# ocrdump -stdout -backupfile week.ocr | diff - OCRDUMPFILE
```

Checking the Integrity of the OCR

Use the `ocrcheck` command to check OCR integrity.

```
# ocrcheck
Status of Oracle Cluster Registry is as follows :
    Version                  :          2
    Total space (kbytes)     :     275980
    Used space (kbytes)      :       2824
    Available space (kbytes) :     273156
    ID                       : 1274772838
    Device/File Name         :  +DATA1
                           Device/File integrity check succeeded
    Device/File Name         :  +DATA2
                           Device/File integrity check succeeded
    Cluster registry integrity check succeeded
    Logical corruption check succeeded
```

ORACLE

Checking the Integrity of the OCR

The `ocrcheck` utility displays the version of OCR's block format, total space available and used space, OCRID, and the OCR locations that you have configured. The `ocrcheck` utility performs a block-by-block checksum operation for all the blocks in all the OCRs that you have configured. It also returns an individual status for each file as well as a result for the overall OCR integrity check.

OCR-Related Tools for Debugging

- OCR tools:
  - ocrdump
  - ocrconfig
  - ocrcheck
  - srvctl
- Logs are generated in the following directory:
  <Grid_Home>/log/<hostname>/client/
- Debugging is controlled through the following file:
  <Grid_Home>/srvm/admin/ocrlog.ini

```
mesg_logging_level = 5
comploglvl="OCRAPI:5 ; OCRSRV:5; OCRCAC:5; OCRMAS:5; OCRCONF:5; OCRRAW:5"
comptrclvl="OCRAPI:5 ; OCRSRV:5; OCRCAC:5; OCRMAS:5; OCRCONF:5; OCRRAW:5"
```

ORACLE

OCR-Related Tools for Debugging

As you have seen, you can use various tools to manipulate the OCR: `ocrdump`, `ocrconfig`, `ocrcheck`, and `srvctl`.

These utilities create log files in <Grid_Home>/log/<hostname>/client/. To change the amount of logging, edit the <Grid_Home>/srvm/admin/ocrlog.ini file. The default logging level is `0`, which basically means minimum logging. When `mesg_logging_level` is set to `0`, which is its default value, only error conditions are logged. You can change this setting to `3` or `5` for detailed logging information.

If that is not enough, you can also change the logging and trace levels for each of the components used to manipulate the OCR. To do that, edit the entries containing `comploglvl` and `comptrclvl` in `ocrlog.ini`.

The slide shows you the three lines you could add to `ocrlog.ini` to turn on additional debugging information for some OCR components. A typical example where you may have to change the `ocrlog.ini` file is in a situation where you get errors while using either the `ocrdump` or `ocrconfig` tool.

OCR-Related Tools for Debugging

Oracle Cluster Registry (OCR) Components

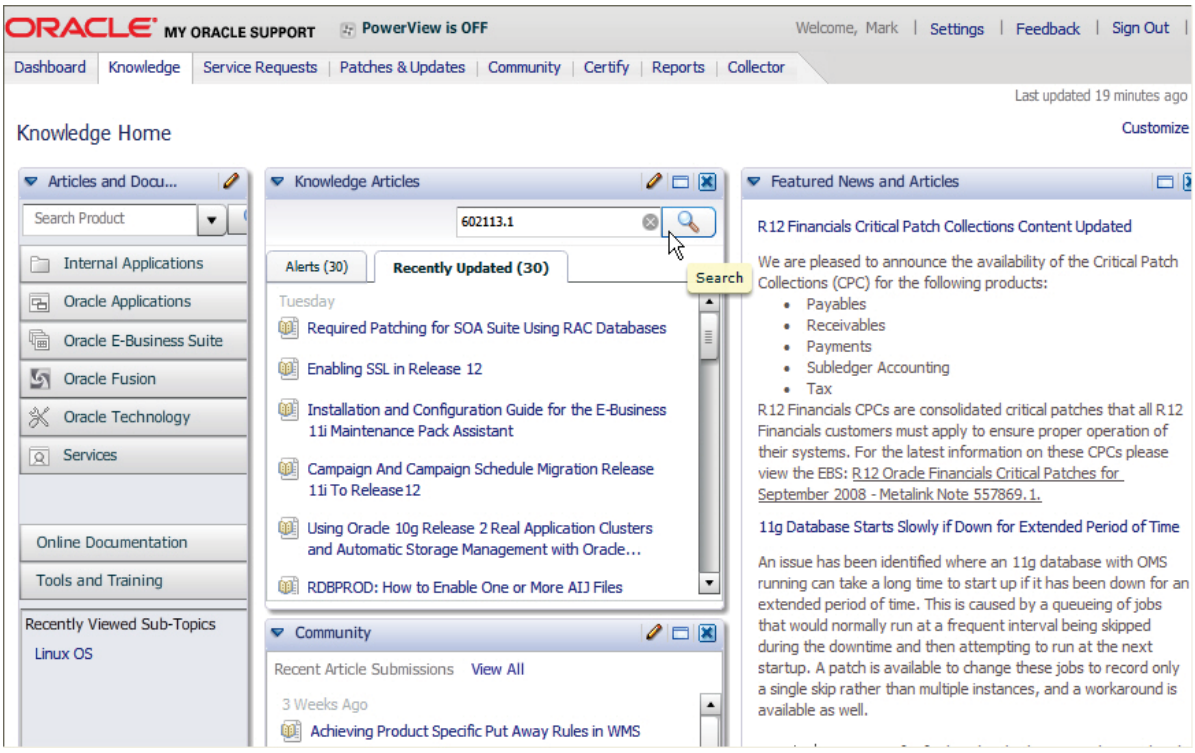
Component Name	Description
OCRAPI	OCR abstraction component
OCRCLI	OCR client component
OCRSRV	OCR server component
OCRMAS	OCR master thread component
OCRMSG	OCR message component
OCRCAC	OCR cache component
OCRRAW	OCR raw device component
OCRUTL	OCR util component
OCROSD	OCR operating system dependent (OSD) layer
OCRASM	OCR ASM component

ORACLE

OCR-Related Tools for Debugging (Continued)

Using an initialization file or the `crsctl set log crs` command, you can debug the OCR components listed in the table above. The components listed can also be used for the Oracle Local Registry (OLR) except for `OCRMAS` and `OCRASM`. You can also use them for the OCR and OLR clients, except for `OCRMAS` and `OCRSRV`. Some of the OCR and OLR clients are `ocrconfig`, `ocrdump`, `ocrcheck`, and so on.

Browsing My Oracle Support Knowledge Articles



Browsing My Oracle Support Knowledge Articles

My Oracle Support is the next generation of Oracle's award-winning support platform integrated with Software Configuration Manager for embedding configuration management capabilities directly into the support processes. Simply by installing the configuration manager and uploading your configuration information, you can enable Oracle's personalized and proactive support capabilities, driven by Oracle's vast knowledge base. This proactive advice includes customized security and product health checks that identify problems and recommend fixes before they impact your system performance. One individual in your organization should be designated as a Oracle support administrator that can create additional access accounts for other members in your organization. After signing in at http://support.oracle.com, knowledge articles may be accessed to assist troubleshooting Oracle Clusterware.

Quiz

Configuration of the `hangcheck-timer` module is required for Grid Infrastructure 11*g* Release 2 clusters.

1. True
2. False

ORACLE

Answer: 2

False. The `hangcheck-timer` module is no longer needed in Oracle Grid Infrastructure 11*g* Release 2 clusters.

Quiz

Which of the following statements regarding `cluvfy` are true?

1. You use it to perform a full stack verification.
2. It uses nonintrusive verification methodology.
3. It works only on clusters employing ASM for shared storage.
4. You can generate fixup scripts with some CVU commands by using the `-fixup` flag.

ORACLE

Answer: 1,2,4

Statements 1, 2, and 4 are correct.

Summary

In this lesson, you should have learned how to:

- Locate Oracle Clusterware log files
- Gather all log files using `diagcollection.pl`
- Enable resource debugging
- Enable component-level debugging
- Enable tracing for Java-based tools
- Troubleshoot the Oracle Cluster Registry (OCR) file

ORACLE

Practice 6 Overview

This practice covers the following topics:

- Working with log files
- Working with `ocrdump`
- Working with `cluvfy`

ORACLE

E

Cloning Oracle Clusterware

Objectives

After completing this lesson, you should be able to:

- Describe the cloning procedure
- Prepare the software for cloning
- Describe the cloning script variables
- Clone Oracle Clusterware to create a new cluster
- Clone Oracle Clusterware to extend an existing cluster
- Examine cloning log files

ORACLE

What Is Cloning?

Cloning is the process of copying an existing Oracle Clusterware installation to a different location. It:

- Requires a successful installation as a baseline
- Can be used to create new clusters
- Cannot be used to remove nodes from the cluster
- Does not perform the operating system prerequisites to an installation
- Is useful to build many clusters in an organization

ORACLE

What Is Cloning?

Cloning is a process that allows the copying of an existing Oracle Clusterware installation to a different location and then updating the copied installation to work in the new environment. The cloned copy can be used to create a new cluster from a successfully installed cluster. To add or delete Oracle Clusterware from nodes in the cluster, use the `addNode.sh` and `rootcrs.pl` scripts. The cloning procedure cannot be used to remove a node from an existing cluster.

The cloning procedure is responsible for the work that would have been done by the Oracle Universal Installer (OUI) utility. It does not automate the prerequisite work that must be done on each node before installing the Oracle software.

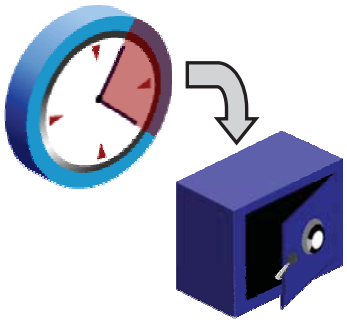
This technique is very useful if a large number of clusters need to be deployed in an organization. If only one or two clusters are being deployed, you should probably use the traditional installation program to perform the installations.

Note: The Oracle Enterprise Manager administrative tool with the Provisioning Pack feature installed has automated wizards to assist with cloning exercises.

Benefits of Cloning Oracle Clusterware

The following are some of the benefits of cloning Oracle Clusterware:

- Can be completed in silent mode from a Secure Shell (SSH) terminal session
- Contains all patches applied to the original installation
- Can be done very quickly
- Is a guaranteed method of repeating the same installation on multiple clusters

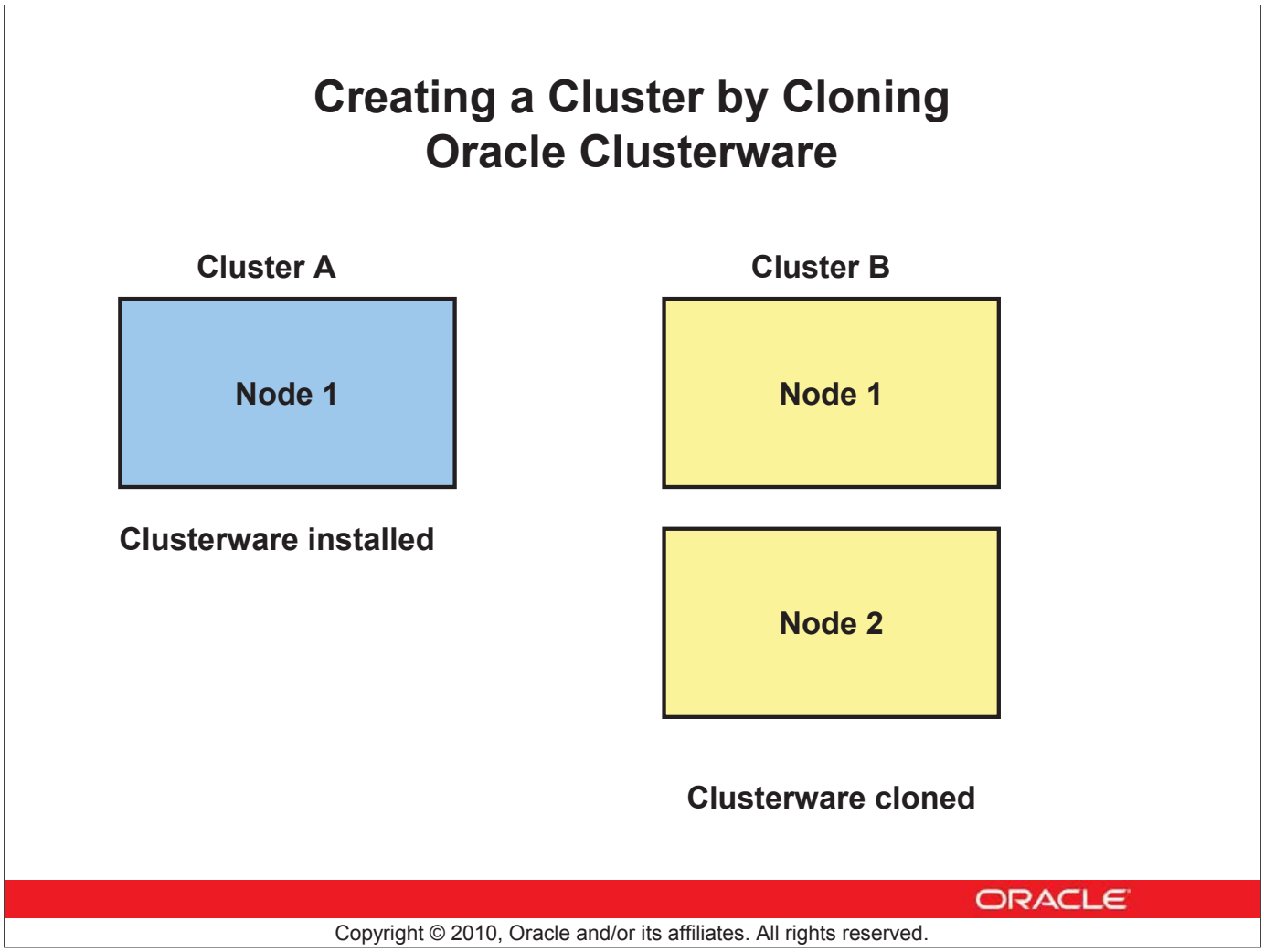


ORACLE

Benefits of Cloning Oracle Clusterware

Using the cloning procedure presented in this lesson has several benefits compared to the traditional Oracle Universal Installer (OUI) installation method. The OUI utility is a graphical program and must be executed from a graphical session. Cloning can be completed in silent mode from a command-line Secure Shell (SSH) terminal session without the need to load a graphical windows system. If the OUI program were to be used to install the software from the original installation media, all patches that have been applied since the first installation would have to be reapplied. The clone technique presented in this lesson includes all successfully applied patches, and can be performed very quickly to a large number of nodes. When the OUI utility performs the copying of files to remote servers, the job is executed serially to one node at a time. With cloning, simultaneous transfers to multiple nodes can be achieved. Finally, the cloning method is a guaranteed way of repeating the same installation on multiple clusters to help avoid human error.

The cloned installation acts the same as the source installation. It can be patched in the future and removed from a cluster if needed using ordinary tools and utilities. Cloning is an excellent way to instantiate test clusters or a development cluster from a successful base installation.



Creating a Cluster by Cloning Oracle Clusterware

This example shows the result when you successfully clone an installed Oracle Clusterware environment to create a cluster. The environment on Node 1 in Cluster A is used as the source, and Cluster B, Nodes 1 and 2 are the destination. The clusterware home is copied from Cluster A, Node 1, to Cluster B, Nodes 1 and 2. When completed, there will be two separate clusters. The OCR and Voting disks are not shared between the two clusters after you successfully create a cluster from a clone.

Preparing the Oracle Clusterware Home for Cloning

The following procedure is used to prepare the Oracle Clusterware home for cloning:

1. Install Oracle Clusterware on the first machine.
   A. Use the Oracle Universal Installer (OUI) GUI interactively.
   B. Install patches that are required (for example, 11.1.0.*n*).
   C. Apply one-off patches, if necessary.
2. Shut down Oracle Clusterware.

```
# crsctl stop crs -wait
```

3. Make a copy of the Oracle Clusterware home.

```
# mkdir /stagecrs
# cp -prf /u01/app/11.2.0/grid /stagecrs
```

ORACLE

Preparing the Oracle Clusterware Home for Cloning

The cloning method requires that an existing, successful installation of Oracle Clusterware be already performed in your organization. Verify that all patch sets and one-off patches have been applied before starting the clone procedure to minimize the amount of work that will have to be performed for the cloning exercise.

To begin the cloning process, start by performing a shutdown of Oracle Clusterware on one of the nodes in the existing cluster with the `crsctl stop crs –wait` command. The other nodes in the existing cluster can remain active. After the prompt is returned from the shutdown command, make a copy of the existing Oracle Clusterware installation into a temporary staging area of your choosing. The disk space requirements in the temporary staging area will be equal to the current size of the existing Oracle Clusterware installation.

The copy of the Oracle Clusterware files will not include files that are outside the main installation directory such as `/etc/oraInst.loc` and the `/etc/oracle` directory. These files will be created later by running various `root` scripts.

Preparing the Oracle Clusterware Home for Cloning

4. Remove files that pertain only to the source node.

```
# cd /stagecrs/grid
# rm -rf /stagecrs/grid/log/<hostname>
# rm -rf root.sh*
# rm -rf gpnp/*
# find . -name '*.ouibak' -exec rm {} \;
# find . -name '*.ouibak.1' -exec rm {} \;
# rm –rf  \
  inventory/ContentsXML/oraclehomeproperties.xml
# cd ./cfgtoollogs
# find . -type f -exec rm -f {} \;
```

5. Create an archive of the source.

```
# cd /stagecrs/grid
# tar –zcvf /tmp/crs111060.tgz .
```

ORACLE

Preparing the Oracle Clusterware Home for Cloning (continued)

Each installation of Oracle Clusterware on local storage devices contains files and directories that are applicable to only that node. These files and directories should be removed before making an archive of the software for cloning. Perform the commands in the slide to remove node-specific information from the copy that was made in step 3. Then create an archive of the Oracle Clusterware copy. An example of using the Linux `tar` command followed by the `compress` command to reduce the file size is shown in the slide. On Windows systems, use the WinZip utility to create a zip file for the archive.

Note: Do not use the Java Archive (JAR) utility to copy and compress the Oracle Clusterware home.

Preparing the Oracle Clusterware Home for Cloning

6. Restart Oracle Clusterware.

```
# crsctl start crs
```

ORACLE

Cloning to Create a New Oracle Clusterware Environment

The following procedure uses cloning to create a new cluster:

1. Prepare the new cluster nodes. (See the lesson titled "Grid Infrastructure Installation" for details.)
   A. Check system requirements.
   B. Check network requirements.
   C. Install the required operating system packages.
   D. Set kernel parameters.
   E. Create groups and users.
   F. Create the required directories.
   G. Configure installation owner shell limits.
   H. Configure SSH and enable user equivalency.
   I. Use the Cluster Verify Utility (`cluvfy`) to check prerequisites.

ORACLE

Cloning to Create a New Oracle Clusterware Environment

The archive of the source files that was made when preparing the Oracle Clusterware home for cloning will become the source files for an installation on the new cluster system and the `clone.pl` script will be used to make it a working environment. These actions do not perform the varied prerequisite tasks that must be performed on the operating system before an Oracle Clusterware installation. The prerequisite tasks are the same as the ones presented in the lesson titled "Grid Infrastructure Installation" in the topic about installing Oracle Clusterware.

One of the main benefits of cloning is rapid instantiation of a new cluster. The prerequisite tasks are manual in nature, but it is suggested that a shell script be developed to automate these tasks. One advantage of the Oracle Enterprise Linux operating system is that an `oracle-validated-1.x.x.rpm` file can be obtained that will perform almost all the prerequisite checks including the installation of missing packages with a single command.

This cloning procedure is used to build a new distinct cluster from an existing cluster. In step 1, prepare the new cluster nodes by performing prerequisite setup and checks.

If `cluvfy` fails to execute because of user equivalence errors, the passphrase needs to be loaded with the following commands before executing `cluvfy`:

```
exec /usr/bin/ssh-agent $SHELL
ssh-add
```

Cloning to Create a New Oracle Clusterware Environment

2. Deploy Oracle Clusterware on each of the destination nodes.
   A. Extract the tar file created earlier.

```
# mkdir -p /u01/app/11.20/grid
# cd /u01/app/11.2.0
# tar -zxvf /tmp/crs111060.tgz
```

   B. Change the ownership of files, and create Oracle Inventory.

```
# chown -R crs:oinstall /u01/app/11.2.0/grid
# mkdir -p /u01/app/oraInventory
# chown grid:oinstall /u01/app/oraInventory
```

   C. Remove any network files from `/u01/app/11.2.0/grid/network/admin`

```
$ rm /u01/app/11.2.0/grid/network/admin/*
```

ORACLE

Cloning to Create a New Oracle Clusterware Environment (continued)

Step 2 is the deployment and extraction of the source archive to the new cluster nodes. If a shared Oracle Cluster home on a Cluster File System (CFS) is not being utilized, extract the source archive to each node's local file system. It is possible to change the operating system owner to a different one from that of the source with recursive `chown` commands as illustrated in the slide. If other Oracle products have been previously installed on the new nodes, the Central Oracle Inventory directory may already exist. It is possible for this directory to be owned by a different user than the Oracle Grid Infrastructure user; however, both should belong to the same primary group `oinstall`. Execute the `preupdate.sh` script on each target node, logged in as `root`. This script will change the ownership of the CRS home and `root`-owned files to the Oracle CRS user.

The `clone.pl` Script

Cloning to a new cluster and cloning to extend an existing cluster both use a PERL script. The `clone.pl` script is used, which:

- Can be used on the command line
- Can be contained in a shell script
- Accepts many parameters as input
- Is invoked by the PERL interpreter

```
# perl <CRS_home>/clone/bin/clone.pl $E01 $E02 $E03
$E04 $C01 $CO2
```

ORACLE

The `clone.pl` Script

The PERL-based `clone.pl` script is used in place of the graphical OUI utility to perform the installation on the new nodes so that they may participate in the existing cluster or become valid nodes in a new cluster. The script can be executed directly on the command line or at a DOS command prompt for Windows platforms. The `clone.pl` script accepts several parameters as input, typed directly on the command line. Because the `clone.pl` script is sensitive to the parameters being passed to it, including the use of braces, single quotation marks, and double quotation marks, it is recommended that a shell script be created to execute the PERL-based `clone.pl` script to input the arguments. This will be easier to rework if there is a syntax error generated. There are a total of seven arguments that can be passed as parameters to the `clone.pl` script. Four of them define environment variables, and the remaining three supply processing options. If your platform does not include a PERL interpreter, you can download one at:

`http://www.perl.org`

The `clone.pl` Environment Variables

The `clone.pl` script accepts four environment variables as input. They are as follows:

Symbol	Variable	Description
E01	`ORACLE_BASE`	The location of the Oracle base directory
E02	`ORACLE_HOME`	The location of the Oracle Grid Infrastructure home. This directory location must exist and must be owned by the Oracle operating system group: oinstall.
E03	`ORACLE_HOME_NAME`	The name of the Oracle Grid Infrastructure home
E04	`INVENTORY_LOCATION`	The location of the Oracle Inventory

ORACLE

The `clone.pl` Environment Variables

The `clone.pl` script accepts four environment variables as command-line arguments providing input. These variables would correlate to some of the questions presented by the OUI utility during an installation. Each variable is associated with a symbol that is used for reference purposes for developing a shell script. The choice of symbol is arbitrary. Each variable is case-sensitive. The variables are as follows:

- **`ORACLE_BASE`:** The location of the Oracle Base directory. A suggested value is `/u01/app/grid`. The `ORACLE_BASE` value should be unique for each software owner.
- **`ORACLE_HOME`:** The location of the Oracle Grid Infrastructure home. This directory location must exist and be owned by the Oracle Grid Infrastructure software owner and the Oracle Inventory group, typically `grid:oinstall`. A suggested value is `/u01/app/<version>/grid`.
- **`ORACLE_HOME_NAME`:** The name of the Oracle Grid Infrastructure home. This is stored in the Oracle inventory and defaults to the name `Ora11g_gridinfrahome1` when performing an installation with OUI. Any name can be selected, but it should be unique in the organization.
- **`INVENTORY_LOCATION`:** The location of the Oracle Inventory. This directory location must exist and must initially be owned by the Oracle operating system group: oinstall. A typical location is /u01/app/oraInventory.

The `clone.pl` Command Options

The `clone.pl` script accepts two required command options as input. They are as follows:

#	Variable	Data Type	Description
C01	`CLUSTER_NODES`	String	This list of short node names for the nodes in the cluster
C02	`LOCAL_NODE`	String	The short name of the local node

ORACLE

The `clone.pl` Command Options

The `clone.pl` script accepts two command options as input. The command options are case-sensitive and are as follows:

- `CLUSTER_NODES`: This list of short node names for the nodes in the cluster
- `LOCAL_NODES`: The short name of the local node

Cloning to Create a New Oracle Clusterware Environment

3. Create a shell script to invoke `clone.pl` supplying input.

```
#!/bin/sh
ORACLE_BASE=/u01/app/oracle
GRID_HOME=/u01/app/11.2.0/grid
THIS_NODE=`hostname -s`

E01=ORACLE_BASE=${ORACLE_BASE}
E02=ORACLE_HOME=${ORACLE_HOME}
E03=ORACLE_HOME_NAME=OraGridHome1
E04=INVENTORY_LOCATION=${ORACLE_BASE}/oraInventory

#C00="-O'-debug'"
C01="-O'\"CLUSTER_NODES={node1,node2}\"'"
C02="-O'\"LOCAL_NODE=${THIS_NODE}\"'"

perl ${GRID_HOME}/clone/bin/clone.pl -silent $E01 $E02
$E03 $E04 $C01 $C02
```

Syntax per bug 8718041

Cloning to Create a New Oracle Clusterware Environment

The `clone.pl` script requires you to provide many setup values to the script when it is executed. You may enter the values interactively on the command line or create a script to supply the input values. By creating a script, you will have the ability to modify it and execute the script a second time if errors exist. The setup values to the `clone.pl` script are case-sensitive and sensitive to the use of braces, single quotation marks, and double quotation marks.

For step 3, create a shell script that invokes `clone.pl` supplying command-line input variables. All the variables should appear on a single line.

Cloning to Create a New Oracle Clusterware Environment

4. Run the script created in Step 3 on each node.

```
$ /tmp/my-clone-script.sh
```

5. Prepare the `crsconfig_params` file.

Cloning to Create a New Oracle Clusterware Environment (continued)

Step 4: Run the script you created in Step 3 as the operating system user that installed Oracle Clusterware. If you do not have a shared Oracle grid infrastructure home run this script on each node. The `clone.pl` command instantiates the `crsconfig_params` file in the next step.

Step 5: Prepare the `/u01/app/11.2.0/grid/install/crsconfig_params` file on all of the nodes in the cluster. You can copy the file from one node to all of the other nodes. More than 50 parameters are named in this file.

Notes: In 11.2.0.1 the `clone.pl` script does not propagate the `crsconfig_params` file.

Cloning to Create a New Oracle Clusterware Environment

6. Run the `orainstRoot.sh` script on each node.

```
# /u01/app/oraInventory/orainstRoot.sh
```

7. Run the `<CRS_home>/root.sh` and `rootcrs.pl` scripts on each node.

```
# /u01/app/11.2.0/grid/root.sh
# /u01/app/11.2.0/grid/perl/bin/perl \
   -I/u01/app/11.2.0/grid/perl/lib \
   -I/u01/app/11.2.0/grid/crs/install \
   /u01/app/11.2.0/grid/crs/install/rootcrs.pl
```

ORACLE

Cloning to Create a New Oracle Clusterware Environment (continued)

When the shell script containing `clone.pl` successfully completes, it is necessary to execute the `orainstRoot.sh` script for step 6 and the `root.sh` script for step 7 on each node, both as the `root` user.

Note

Step 4 must be run to completion before you start Step 5. Similarly, Step 5 must be run to completion before you start Step 6.

You can perform Step 4, Step 5, and Step 6 simultaneously on different nodes. Step 6 must be complete on all nodes before you can run Step 7.

Step 7: Ensure that the `root.sh` and `rootcrs.pl` scripts have completed on the first node before running them on the second node and subsequent nodes.

Cloning to Create a New Oracle Clusterware Environment

8. Run the configuration assistants on each new node.

```
$ /u01/app/11.2.0/grid/bin/netca \
    /orahome /u01/app/11.2.0/grid \
    /orahnam OraGridHome1 /instype typical \
    /inscomp client,oraclenet,javavm,server\
    /insprtcl tcp /cfg local \
    /authadp NO_VALUE \
    /responseFile \
 /u01/app/11.2.0/grid/network/install/netca_typ.rsp \
    /silent
```

9. Run the `cluvfy` utility to validate the installation.

```
$ cluvfy stage -post crsinst -n all -verbose
```

ORACLE

Cloning to Create a New Oracle Clusterware Environment (continued)

Step 8: You should execute the command shown in the slide, on the first node.

If you are using ASM, then run the following command:

```
$ /u01/app/11.2.0/grid/bin/asmca -silent -postConfigureASM \
 -sysAsmPassword oracle -asmsnmpPassword oracle
```

If you a plan to run a pre-11*g* Release 2 (11.2) database on this cluster, then you should run `oifcfg` as described in the Oracle Database 11*g* Release 2 (11.2) documentation.

To use IPMI, use the `crsctl` command to configure IPMI on each node:

```
# crsctl set css ipmiaddr ip_address
```

At this point, Oracle Clusterware is fully installed and configured to operate in the new cluster. The `cluvfy` utility can be invoked in the post-CRS installation stage to verify the installation with the following syntax:

```
cluvfy stage -post crsinst -n all -verbose
```

If `cluvfy` fails to execute because of user equivalence errors, the passphrase needs to be loaded with the following commands before executing `cluvfy` again:

```
exec /usr/bin/ssh-agent $SHELL
ssh-add
```

Log Files Generated During Cloning

The following log files are generated during cloning to assist with troubleshooting failures.

- Detailed log of the actions that occur during the OUI part:

```
/u01/app/oraInventory/logs/cloneActions<timestamp>.log
```

- Information about errors that occur when OUI is running:

```
/u01/app/oraInventory/logs/oraInstall<timestamp>.err
```

- Other miscellaneous messages generated by OUI:

```
/u01/app/oraInventory/logs/oraInstall<timestamp>.out
```

ORACLE

Log Files Generated During Cloning

Several log files are generated when the `clone.pl` script is executed and are useful in diagnosing any errors that may occur. For a detailed log of the actions that occur during the OUI part of the cloning, examine the log file:

```
<Central_Inventory>logs/cloneActions/<timestamp>.log
```

If errors occurred during the OUI portion of the cloning process, examine the log file:

```
<Central_Inventory>logs/oraInstall/<timestamp>.err
```

Other miscellaneous messages generated by OUI can be found in the output file:

```
<Central_Inventory>logs/oraInstall/<timestamp>.out
```

Log Files Generated During Cloning

The following log files are generated during cloning to assist with troubleshooting failures:

- Detailed log of the actions that occur before cloning as well as during cloning operations:

```
/u01/app/grid/clone/logs/clone-<timestamp>.log
```

- Information about errors that occur before cloning as well as during cloning operations:

```
/u01/app/grid/clone/logs/error-<timestamp>.log
```

ORACLE

Log Files Generated During Cloning (continued)

Several log files are generated when the `clone.pl` script is executed and are useful in diagnosing errors that may occur. For a detailed log of the actions that occur before cloning as well as during the cloning process, examine the log file:

```
<CRS home>/clone/logs/clone-<timestamp>.log
```

For a detailed log of the errors that occur before cloning as well as during the cloning process, examine the log file:

```
<CRS home>/clone/logs/error-<timestamp>.log
```

Cloning to Extend Oracle Clusterware to More Nodes

The procedure is very similar to cloning to create new clusters.

1. Prepare the new cluster nodes. (See the lesson titled "Oracle Clusterware Installation" for details. Suggest that this be made into a shell script for reuse.)
   A. Check system requirements.
   B. Check network requirements.
   C. Install the required operating system packages.
   D. Set kernel parameters.
   E. Create groups and users.
   F. Create the required directories.
   G. Configure installation owner shell limits.
   H. Configure SSH and enable user equivalency.
   I. Use the `cluvfy` utility to check prerequisites.

ORACLE

Cloning to Extend Oracle Clusterware to More Nodes

The cloning procedure using the `clone.pl` script can also be used to extend Oracle Clusterware to more nodes within the same cluster with steps similar to the procedure for creating a new cluster. The first step shown here is identical to the first step performed when cloning to create a new cluster. These steps differ depending on the operating system used.

When you configure Secure Shell (SSH) and enable user equivalency, remember that the `authorized_keys` and `known_hosts` files exist on each node in the cluster. Therefore, it will be necessary to update these files on the existing nodes with information about the new nodes that Oracle Clusterware will be extended to.

Note: Not supported in 11.2.0.1

Cloning to Extend Oracle Clusterware to More Nodes

2. Deploy Oracle Clusterware on the destination nodes.
   A. Extract the tar file created earlier.

```
# mkdir -p /u01/app/crs
# cd /u01/app/crs
# tar -zxvf /tmp/crs111060.tgz
```

   B. Change the ownership of files and create Oracle Inventory.

```
# chown -R crs:oinstall /u01/app/crs
# mkdir -p /u01/app/oraInventory
# chown crs:oinstall /u01/app/oraInventory
```

   C. Run the `preupdate.sh` script on each target node.

```
# /u01/app/crs/install/preupdate.sh \
-crshome /u01/app/crs -crsuser crs -noshutdown
```

ORACLE

Cloning to Extend Oracle Clusterware to More Nodes (continued)

Step 2 is the deployment and extraction of the source archive to the new cluster nodes. If a shared Oracle Cluster home on a CFS is not being used, extract the source archive to each node's local file system. Because this node will participate with the existing nodes in a cluster, it is not possible to use different account names for the Oracle Clusterware software owner. If other Oracle products have been previously installed on the new nodes, the Central Oracle Inventory directory may already exist. It is possible for this directory to be owned by a different user than the Oracle Clusterware user; however, both should belong to the same primary group `oinstall`. Execute the `preupdate.sh` script on each target node, logged in as `root`. This script will change the ownership of the CRS home and `root`-owned files to the Oracle CRS user if needed.

Cloning to Extend Oracle Clusterware to More Nodes

3. Create a shell script to invoke `clone.pl` supplying input.

```
#!/bin/sh
# /tmp/my-clone-script.sh
E01=ORACLE_BASE=/u01/app
E02=ORACLE_HOME=/u01/app/crs
E03=ORACLE_HOME_NAME=OraCrs11g
C01="-O'sl_tableList={node3:node3-priv:node3-vip:N:Y}'"
C02="-O'INVENTORY_LOCATION=/u01/app/oraInventory'"
C03="-O'-noConfig'"

perl /u01/app/crs/clone/bin/clone.pl $E01 $E02 $E03 $C01 $C02 $C03
```

4. Run the shell script created in step 3 on each new node.

```
$ /tmp/my-clone-script.sh
```

ORACLE

Cloning to Extend Oracle Clusterware to More Nodes (continued)

For step 3, the quantity of input variables to the `clone.pl` procedure is greatly reduced because the existing cluster has already defined many of the settings that are needed. Creating a shell script to provide these input values is still recommended. For step 4, run the shell script that you developed to invoke the `clone.pl` script.

Cloning to Extend Oracle Clusterware to More Nodes

5. Run the `orainstRoot.sh` script on each new node.

```
# /u01/app/oraInventory/orainstRoot.sh
```

6. Run the `addNode` script on the source node.

```
$ /u01/app/crs/oui/bin/addNode.sh -silent \
"CLUSTER_NEW_NODES={new_nodes}" \
"CLUSTER_NEW_PRIVATE_NODE_NAMES={new_node-priv}" \
"CLUSTER_NEW_VIRTUAL_HOSTNAMES={new_node-vip}" \
-noCopy
```

7. Run the `rootaddnode.sh` script on the source node.

```
# /u01/app/crs/install/rootaddnode.sh
```


Cloning to Extend Oracle Clusterware to More Nodes (continued)

For step 5, run the `orainstRoot.sh` script on each new node as the `root` user. For step 6, run the `addNode.sh` script on the source node, not on the destination node. Because the `clone.pl` scripts have already been run on the new nodes, this script only updates the inventories of the existing nodes. For step 7, again on the source node, run the `rootaddnode.sh` script as `root` to instantiate the node.

Cloning to Extend Oracle Clusterware to More Nodes

8. Run the `<CRS_home>/root.sh` script on the new node.

```
# /u01/app/crs/root.sh
```

9. Run the configuration assistants on each node that is listed in the `configToolAllCommands` file.

```
$ cat /u01/app/crs/cfgtoollogs/configToolAllCommands
```

```
$ onsconfig add_config node3:6501
```

10. Run `cluvfy` to validate the installation.

```
$ cluvfy stage -post crsinst -n all -verbose
```

ORACLE

Cloning to Extend Oracle Clusterware to More Nodes (continued)

Next, in step 8, run the `root.sh` script on each new node joining the cluster as the `root` user. For step 9, additional commands for the configuration assistants need to be run on the new nodes joining the cluster as the Oracle Clusterware software owner. The list of commands can be found in the `<CRS home>/cfgtoollogs/configToolAllCommands` file. Finally, for step 10, which is the last step, run `cluvfy` to verify the success of the Oracle Clusterware installation.

Quiz

An Oracle Clusterware home that was created with cloning techniques can be used as the source for additional cloning exercises.

1. True
2. False

ORACLE

Answer: 1

Quiz

Which scripting language is used for the cloning script?

1. Java
2. PERL
3. Shell
4. Python

ORACLE

Answer: 2

Summary

In this lesson, you should have learned how to:

- Describe the cloning process
- Describe the `clone.pl` script and its variables
- Perform a clone of Oracle Clusterware to a new cluster
- Extend an existing cluster by cloning

ORACLE