



Integrated Cloud Applications & Platform Services

Oracle Database 12c R2: Clusterware Administration

Student Guide

D81246GC20

Edition 2.0 | March 2018 | D103313

Learn more from Oracle University at education.oracle.com



Author

Sean Kim

**Technical Contributors
and Reviewers**

Sean Kim
Jerry Lee
Joel Goodman
Frank Fu
James Womack

Graphic Editor

Seema Bopaiah

Publishers

Pavithran Adka
Asief Baig
Raghunath M

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Disclaimer

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

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

Restricted Rights Notice

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

U.S. GOVERNMENT RIGHTS

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

Trademark Notice

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

Contents

1 Introduction to Clusterware

Objectives 1-2
Cluster 1-3
Clusterware 1-4
Oracle Clusterware 1-5
Clusterware Architecture and Cluster Services 1-6
Goals for Oracle Clusterware 1-7
Oracle Clusterware Fencing 1-8
Cluster Time Synchronization 1-10
Network Resource Management 1-11
Oracle Clusterware Operating System Requirements 1-12
Oracle Clusterware Networking 1-13
IP Addresses for Public Networks 1-15
Private Network IPv6 Support 1-16
Grid Naming Service (GNS) 1-17
Grid Naming Service Configuration Options 1-18
Shared GNS Across Multiple Clusters 1-20
Highly Available Grid Naming Service 1-21
Configuring Highly Available GNS 1-23
Single-Client Access Name 1-24
Quiz 1-26
Summary 1-28

2 Oracle Clusterware Architecture

Objectives 2-2
Oracle Clusterware Technology Stack 2-3
Cluster Ready Services Technology Stack 2-4
OHAS Technology Stack 2-6
Clusterware Component Processes and Services 2-8
Oracle Clusterware Repository (OCR) 2-9
CSS Voting Disk Function 2-10
Voting Disk Considerations 2-11
Oracle Local Registry and High Availability 2-12
Oracle Clusterware Initialization 2-13
Clusterware Startup Details 2-15

| | |
|--|------|
| Clusterware Startup: OHASD orarootagent | 2-17 |
| Clusterware Startup Details: CRSD orarootagent | 2-19 |
| Clusterware Startup Details: CRSD oraagent | 2-20 |
| Clusterware Startup Details: OHASD oraagent | 2-21 |
| Controlling Oracle Clusterware | 2-22 |
| Verifying the Status of Oracle Clusterware | 2-23 |
| Viewing the High Availability Services Stack | 2-24 |
| GPnP Architecture: Overview | 2-25 |
| How GPnP Works: Cluster Node Startup | 2-27 |
| Client Database Connections | 2-28 |
| Quiz | 2-29 |
| Summary | 2-30 |
| Practice 2: Overview | 2-31 |

3 Cluster Configuration Options

| | |
|--|------|
| Objectives | 3-2 |
| Cluster Configuration Options | 3-3 |
| Oracle Standalone Clusters | 3-4 |
| Oracle Cluster Domain | 3-6 |
| Oracle Domain Services Cluster | 3-7 |
| Oracle Member Clusters | 3-8 |
| Oracle Member Cluster for Oracle Databases | 3-9 |
| Oracle Member Cluster for Applications | 3-10 |
| Member Cluster Manifest File for Member Clusters | 3-11 |
| Oracle Extended Clusters | 3-14 |
| Option 1: Configure an Oracle Extended Cluster | 3-15 |
| Assign Failure Groups to Sites | 3-16 |
| Option 2: Configure Oracle Extended Clusters | 3-17 |
| Assign Failure Groups to Sites Using ASMCA | 3-19 |
| Quiz | 3-20 |
| Summary | 3-21 |

4 Grid Infrastructure Preinstallation Tasks

| | |
|---|-----|
| Objectives | 4-2 |
| Preinstallation Planning | 4-3 |
| Shared Storage Planning for Grid Infrastructure and RAC | 4-4 |
| Using a Shared File System with Grid Infrastructure | 4-5 |
| Logical Volume Managers and Grid Infrastructure | 4-6 |
| Managing Voting Disks in ASM | 4-7 |
| Sizing Storage for Oracle Standalone Cluster | 4-8 |
| GIMR Configuration Details | 4-9 |

| | |
|---|------|
| Quiz | 4-10 |
| Grid Infrastructure Preinstallation Tasks | 4-11 |
| Oracle Grid Infrastructure 12c Installation | 4-12 |
| General Server Minimum Requirements | 4-13 |
| Checking System Requirements | 4-14 |
| Enabling the Name Service Cache Daemon (nscd) | 4-15 |
| Setting the Disk I/O Scheduler on Linux | 4-16 |
| Cluster Name and SCAN Requirements | 4-17 |
| Checking Network Requirements | 4-18 |
| IP Address Requirements with GNS | 4-20 |
| IP Address Requirements for Static Configuration | 4-21 |
| Broadcast and Multicast Requirements | 4-23 |
| Private Interconnect Network Requirements | 4-24 |
| Interconnect NIC Guidelines | 4-25 |
| Private Interconnect Redundant Network Requirements | 4-26 |
| Interconnect Link Aggregation: Single Switch | 4-27 |
| Interconnect Link Aggregation: Multiswitch | 4-29 |
| Additional Interconnect Guidelines | 4-30 |
| Cluster Time Synchronization | 4-31 |
| Software Requirements (Kernel) | 4-33 |
| Software Requirements: Packages | 4-34 |
| Oracle Linux with the Unbreakable Enterprise Kernel | 4-37 |
| Zero-Downtime Kernel Updates with Ksplice | 4-38 |
| Oracle Preinstallation RPM | 4-39 |
| Installing the cvuqdisk RPM for Linux | 4-40 |
| Creating Groups and Users | 4-42 |
| Creating Groups, Users, and Paths | 4-43 |
| Shell Settings for the Grid Infrastructure User | 4-44 |
| Determining Root Script Execution Plan | 4-45 |
| Quiz | 4-46 |
| Summary | 4-47 |
| Practice 4: Overview | 4-48 |

5 Grid Infrastructure Installation

| | |
|----------------------------------|-----|
| Objectives | 5-2 |
| Installing Grid Infrastructure | 5-3 |
| Choosing a Cluster Configuration | 5-4 |
| Grid Plug and Play Support | 5-5 |
| Configuring Shared GNS | 5-7 |
| Cluster Node Information | 5-8 |
| Specify Network Interface Usage | 5-9 |

| | |
|--|------|
| Storage Option Information | 5-10 |
| Create ASM Disk Group | 5-11 |
| Create ASM Disk Group: Specify Failure Groups | 5-13 |
| Specify ASM Password | 5-14 |
| Failure Isolation Support with IPMI | 5-15 |
| Specify Management Options | 5-17 |
| Privileged Operating System Groups | 5-18 |
| Specify Installation Location | 5-19 |
| Create Inventory | 5-20 |
| Root Script Execution Configuration | 5-21 |
| Perform Prerequisite Checks | 5-22 |
| Install Product | 5-23 |
| Verifying the Grid Infrastructure Installation | 5-24 |
| Understanding Offline Processes | 5-25 |
| Check ASM Function for Oracle Clusterware Files | 5-26 |
| Create a Fast Recovery Area Disk Group | 5-27 |
| Modifying Oracle Clusterware Binaries After Installation | 5-29 |
| Unconfiguring Oracle Clusterware Without Removing Binaries | 5-31 |
| Quiz | 5-32 |
| Summary | 5-33 |
| Practice 5: Overview | 5-34 |

6 Managing Cluster Nodes

| | |
|---|------|
| Objectives | 6-2 |
| Adding a Cluster Node | 6-3 |
| Using Oracle Grid Infrastructure Installer: Example | 6-5 |
| Prerequisite Steps for Running addnode.sh | 6-6 |
| Prerequisite Steps for Running addNode.sh | 6-7 |
| Adding a Node with addnode.sh | 6-8 |
| Completing OUI Silent Node Addition | 6-10 |
| Adding a Node to a Cluster on Windows Systems | 6-11 |
| Deleting a Node from the Cluster | 6-13 |
| Deleting a Node from a Windows-Based Cluster | 6-18 |
| Quiz | 6-19 |
| Summary | 6-20 |
| Practice 6: Overview | 6-21 |

7 Traditional Clusterware Management

| | |
|-----------------------------|-----|
| Objectives | 7-2 |
| Managing Oracle Clusterware | 7-3 |
| Role-Separated Management | 7-4 |

| | |
|--|------|
| Configuring Horizontal Role Separation | 7-6 |
| Controlling Oracle Clusterware | 7-8 |
| Verifying the Status of Oracle Clusterware | 7-9 |
| Determining the Location of Oracle Clusterware Configuration Files | 7-10 |
| Checking the Integrity of Oracle Clusterware Configuration Files | 7-11 |
| Locating the OCR Automatic Backups | 7-12 |
| Changing the Automatic OCR Backup Location | 7-13 |
| Considerations of Managing OCR and Voting Disks | 7-14 |
| Adding, Replacing, and Repairing OCR Locations | 7-15 |
| Removing an Oracle Cluster Registry Location | 7-16 |
| Migrating OCR Locations to ASM | 7-17 |
| Migrating OCR from ASM to Other Shared Storage | 7-18 |
| Performing Manual OCR Backups | 7-19 |
| Restoring the OCR on Linux or UNIX Systems Case 1: OCR in a Non-ASM Storage | 7-20 |
| Restoring the OCR on Linux or UNIX Systems Case 2: OCR in an ASM Disk Group | 7-21 |
| Restoring the OCR on Linux or UNIX Systems | 7-23 |
| Backing Up and Recovering the Voting Disk | 7-24 |
| Adding, Deleting, or Migrating Voting Disks | 7-25 |
| Restoring Voting Disks | 7-26 |
| Restoring Voting Disks Case 1: Voting Disks in a Non-ASM Storage | 7-27 |
| Restoring Voting Disks Case 2: Voting Disks in an ASM Disk Group | 7-28 |
| Oracle Local Registry | 7-29 |
| Oracle Interface Configuration Tool: oifcfg | 7-31 |
| Determining the Current Network Settings | 7-32 |
| Configuring Redundant Interconnect Usage Using OIFCFG | 7-33 |
| Changing the Virtual IP Addresses Using SRVCTL | 7-34 |
| Changing the Interconnect Adapter Using OIFCFG | 7-37 |
| Managing SCAN VIP and SCAN Listener Resources | 7-39 |
| SCAN Listeners and Valid Node Checking | 7-43 |
| What-If Command Evaluation | 7-44 |
| Performing What-If Command Evaluation on Application Resources with CRSCTL | 7-45 |
| Performing What-If Command Evaluation on Oracle Clusterware Resources with CRSCTL | 7-46 |
| Formatting the Output for What-If Command Evaluation on Oracle Clusterware Resources | 7-47 |
| Performing What-If Command Evaluation with SRVCTL | 7-48 |
| Evaluating Failure Consequences with SRVCTL | 7-49 |
| Reasoned Command Evaluation (Why-If) | 7-50 |

Why-If: Managing Servers, Server Pools, and Policies 7-51
Quiz 7-52
Summary 7-55
Practice 7: Overview 7-56

8 Policy-Based Cluster and Capacity Management

Objectives 8-2
Policy-Based Cluster Management Enhancements: Overview 8-3
Server Pools 8-4
Server Pools and Policy-Based Management 8-5
Server Pool Attributes 8-6
Server Pool Attribute Considerations 8-8
GENERIC and FREE Server Pools 8-9
Assignment of Servers to Server Pools 8-11
Creating Server Pools with crsctl and srvctl 8-12
Managing Server Pools with srvctl and crsctl 8-13
Moving Servers Between Server Pools 8-14
Managing Server Pools Using Default Attributes 8-15
Server State Attributes 8-16
Server Categorization: Overview 8-18
Server Categorization 8-19
Administering Server Categorization: Server Attributes 8-20
Administering Server Categorization: Server Categories 8-21
Administering Server Categorization: Server Pools 8-23
Policy Set: Overview 8-24
Policy-Based Cluster Management and QoS Management 8-26
Viewing the Policy Set 8-27
Configuring a User-Defined Policy Set: Method 1 8-28
Configuring a User-Defined Policy Set: Method 2 8-29
Modifying a User-Defined Policy Set 8-30
Activating a User-Defined Policy 8-31
Load-Aware Resource Placement 8-32
Server Weight-Based Node Eviction 8-33
Assigning Weight to Servers and Resources 8-34
Quiz 8-35
Summary 8-38
Practice 8 Overview: Using Policy-Based Cluster Management 8-39

9 Upgrading and Patching Grid Infrastructure

Objectives 9-2
Clusterware Upgrading and Patching: Overview 9-3

| | |
|--|------|
| Oracle Grid Infrastructure Upgrade | 9-4 |
| Options for Oracle Grid Infrastructure Upgrades | 9-5 |
| Pre-Upgrade Tasks | 9-6 |
| Moving Oracle Clusterware Files to Oracle ASM | 9-7 |
| Using CVU to Validate Readiness for Clusterware Upgrades | 9-8 |
| Understanding Rolling Upgrades Using Batches | 9-9 |
| Performing a Rolling Upgrade from an Earlier Release | 9-11 |
| Completing a Clusterware Upgrade When Nodes Become Unreachable | 9-14 |
| Deinstalling the Old Oracle Clusterware Installation | 9-15 |
| Patch and Patch Set: Overview | 9-16 |
| Types of Patches | 9-17 |
| Obtaining Oracle RAC Patch Sets | 9-18 |
| Obtaining Oracle Clusterware Patches | 9-20 |
| Downloading Patches | 9-22 |
| Grid Infrastructure Patching methods | 9-23 |
| Rolling Patches | 9-25 |
| Checking Software Versions | 9-26 |
| OPatch: Overview | 9-27 |
| OPatch: General Usage | 9-28 |
| Before Patching with OPatch | 9-29 |
| Installing a Patch Manually Using OPatch | 9-30 |
| OPatch Automation | 9-32 |
| Installing a Patch Automatically Using OPatchAuto | 9-33 |
| OPatch Log and Trace Files | 9-35 |
| Queryable Patch Inventory | 9-36 |
| Alternative Methods of Patching | 9-37 |
| Quiz | 9-38 |
| Summary | 9-40 |

10 Monitoring and Troubleshooting Oracle Clusterware

| | |
|---|-------|
| Objectives | 10-2 |
| Lesson Agenda | 10-3 |
| “Golden Rule” in Debugging Oracle Clusterware | 10-4 |
| Oracle Autonomous Health Framework | 10-5 |
| Cluster Verify Utility (CVU) | 10-7 |
| Clusterware resource (ora.cvu) | 10-8 |
| CVU Heath Check Report: Example | 10-10 |
| Cluster Verify Components | 10-11 |
| Cluster Verify Output: Example | 10-12 |
| Cluster Health Monitor (CHM) | 10-13 |
| oclomon Utility | 10-14 |

| | |
|---|-------|
| clomon dumpnodeview Command | 10-15 |
| oclomon dumpnodeview Command | 10-16 |
| oclomon manage Command | 10-17 |
| Cluster Health Monitor (CHM) Enhancements | 10-18 |
| Cluster Health Advisor (CHA) | 10-19 |
| Cluster Health Advisor: Overview | 10-20 |
| Oracle Cluster Health Advisor Architecture | 10-21 |
| Using the CHA Command Line Interface chactl | 10-22 |
| Managing the CHA Models: Defining “normal” | 10-23 |
| CHA Key Performance and Workload Indicators | 10-24 |
| Using chactl query to View Problems and Diagnosis | 10-25 |
| Managing the CHA Repository | 10-26 |
| Trace File Analyzer (TFA) Collector | 10-28 |
| TFA Collector Utility | 10-29 |
| TFA Collector Analysis | 10-30 |
| TFA Collector Repository | 10-31 |
| Managing ADR Logs by Using tfactl managelogs | 10-32 |
| Lesson Agenda | 10-33 |
| Cluster Resource Activity Log (CALOG) | 10-34 |
| Querying and Managing the CALOG | 10-35 |
| Lesson Agenda | 10-36 |
| ADR Directory Structure | 10-37 |
| Files in the Trace Directory | 10-38 |
| Clusterware Trace Files | 10-39 |
| The Oracle Clusterware Alert Log | 10-40 |
| Incident Trace Files | 10-41 |
| Other Diagnostic Data | 10-42 |
| Lesson Agenda | 10-43 |
| Node Eviction: Overview | 10-44 |
| Rebootless Node Eviction: Example | 10-45 |
| Processes Roles For Node Reboots | 10-46 |
| Reboot Advisory in clusterware alert.log | 10-47 |
| Other Log & Trace Files to Review | 10-50 |
| Possible Troubleshooting Scenario: Example | 10-52 |
| Quiz | 10-53 |
| Summary | 10-56 |
| Practice 10: Overview | 10-57 |

11 Making Applications Highly Available with Oracle Clusterware

| | |
|---|------|
| Objectives | 11-2 |
| Oracle Clusterware High Availability (HA) | 11-3 |

| | |
|---|-------|
| Oracle Clusterware HA Components | 11-4 |
| Clusterware Resource Modeling | 11-5 |
| Agents | 11-6 |
| Action Scripts | 11-7 |
| Resource Types | 11-8 |
| Adding Resource Types | 11-10 |
| Adding a Resource Type with EM | 11-11 |
| Using Clusterware to Enable High Availability | 11-12 |
| Resource Attributes | 11-14 |
| Resource States | 11-18 |
| Resource Dependencies | 11-19 |
| Start Dependencies | 11-20 |
| Stop Dependencies | 11-22 |
| Creating a Clusterware Managed Application VIP | 11-23 |
| Creating an Application VIP Using EM | 11-25 |
| Deciding on a Deployment Scheme | 11-26 |
| Registering a Resource | 11-27 |
| Registering a Resource: Example | 11-28 |
| Adding Resources with EM | 11-29 |
| Managing Resources with crsctl | 11-32 |
| Managing Clusterware Resources with EM | 11-34 |
| Clusterware Resource Groups | 11-35 |
| Resource Group: Overview | 11-36 |
| Automatic Resource Groups | 11-37 |
| Resource Group Privileges | 11-38 |
| Resource Group Dependencies | 11-39 |
| Resource Group Dependency Types and Modifiers | 11-40 |
| Failure and Recovery of Critical Resources | 11-41 |
| Failure and Recovery of Non-Critical Resources | 11-42 |
| Resource Group Types | 11-43 |
| Using Resource Groups | 11-44 |
| HA Events: ONS and FAN | 11-45 |
| Managing Oracle Notification Server with srvctl | 11-46 |
| Quiz | 11-47 |
| Summary | 11-50 |
| Practice 11: Overview | 11-51 |

Unauthorized reproduction or distribution prohibited. Copyright© 2019, Oracle and/or its affiliates.

GANG LIU (gangl@baylorhealth.edu) has a non-transferable license
to use this Student Guide.

Introduction to Clusterware



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Explain the principles and purposes of clusters
- Describe cluster hardware best practices
- Describe the Oracle Clusterware architecture

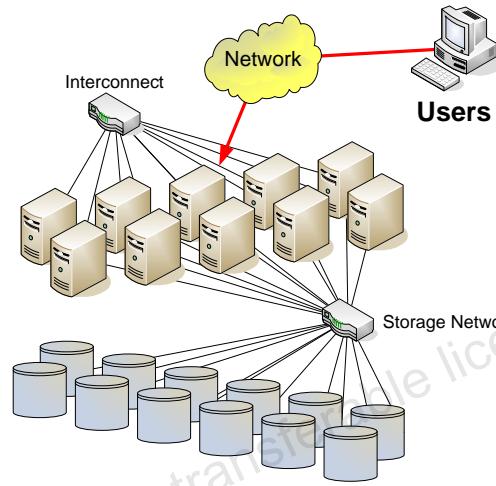


ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Cluster

- A group of independent, but interconnected, computers that act as a single system
- Usually deployed to increase availability and performance or to balance a dynamically changing workload



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

A cluster consists of a group of independent but interconnected computers whose combined resources can be applied to a processing task. A common cluster feature is that it should appear to an application as though it were a single server. Most cluster architectures use a dedicated network (cluster interconnect) for communication and coordination between cluster nodes.

A common cluster architecture for data-intensive transactions and computations is built around shared disk storage. Shared-nothing clusters use an alternative architecture where storage is not shared and data must be either replicated or segmented across the cluster. Shared-nothing clusters are commonly used for workloads that can be easily and predictably divided into small units that can be spread across the cluster in parallel. Shared disk clusters can perform these tasks but also offer increased flexibility for varying workloads. Load-balancing clusters allow a single application to balance its workload across the cluster. Alternatively, in a failover cluster, some nodes can be designated as the primary host for an application, whereas others act as the primary host for different applications. In a failover cluster, the failure of a node requires that the applications it supports be moved to a surviving node. Load-balancing clusters can provide failover capabilities but they can also run a single application across multiple nodes providing greater flexibility for different workload requirements. Oracle supports a shared disk cluster architecture providing load-balancing and failover capabilities. In an Oracle cluster, all nodes must share the same processor architecture and run the same operating system. With the release of Oracle Database 12c, Flex ASM allows nodes in the cluster access to shared storage indirectly through an ASM instance on another node in the cluster.

Clusterware

- Clusterware is a software that provides various interfaces and services for a cluster.
- Typically, this includes capabilities that:
 - Allow the cluster to be managed as a whole
 - Protect the integrity of the cluster
 - Maintain a registry of resources across the cluster
 - Deal with changes to the cluster
 - Provide a common view of resources

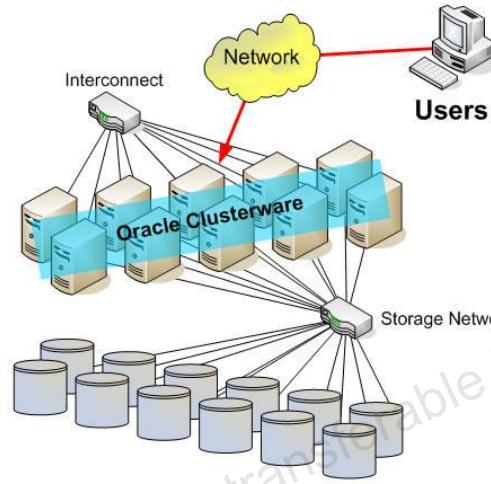


Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware

Oracle Clusterware is:

- A key part of Oracle Grid Infrastructure
- Integrated with Oracle Automatic Storage Management (ASM)
- The basis for Oracle ASM Cluster File System (ACFS)
- A foundation for Oracle Real Application Clusters (RAC)
- A generalized cluster infrastructure for all kinds of applications



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware is a key part of Oracle Grid Infrastructure, which also includes Automatic Storage Management (ASM) and the Oracle ASM Cluster File System. Oracle Clusterware can use ASM for all the shared files required by the cluster. Oracle Clusterware is also a foundation for the ASM Cluster File System, a generalized cluster file system that can be used for most file-based data such as documents, spreadsheets, and reports.

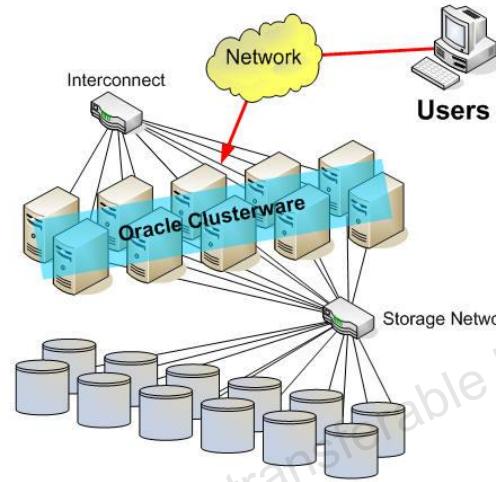
The combination of Oracle Clusterware, ASM, and ASM Cluster File System (ACFS) provides administrators with a unified cluster solution that is not only the foundation for the RAC database, but can also be applied to all kinds of other applications. Oracle Clusterware also manages resources, such as virtual IP (VIP) addresses, databases, listeners, services, and so on.

Using Oracle Clusterware eliminates the need for proprietary vendor clusterware and provides the benefit of using only Oracle software. Oracle provides an entire software solution, including everything from disk management with Oracle Automatic Storage Management (Oracle ASM) to data management with Oracle Database and Oracle RAC. In addition, Oracle Database features, such as Oracle Services, provide advanced functionality when used with the underlying Oracle Clusterware high availability framework.

With the introduction of Oracle 12c Flex Clusters, pure shared disk clusters are not the only type of clustered hardware supported. The architecture has become hybrid with the introduction of hub and leaf nodes.

Clusterware Architecture and Cluster Services

- Shared disk cluster architecture supporting application load balancing and failover
- Services include:
 - Cluster management
 - Node monitoring
 - Event services
 - Time synchronization
 - Network management
 - High availability
 - Cluster Interconnect Link Aggregation (HAIP)



ORACLE®

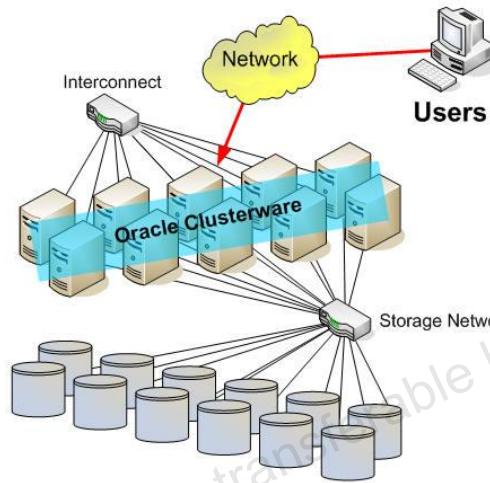
Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware provides a complete set of cluster services to support the shared disk, load-balancing cluster architecture of the Oracle Real Application Cluster (RAC) database. Oracle Clusterware can also be used to provide failover clustering services for single-instance Oracle databases and other applications. The services provided by Oracle Clusterware include:

- Cluster management, which allows cluster services and application resources to be monitored and managed from any node in the cluster
- Node monitoring, which provides real-time information regarding which nodes are currently available and the resources they support. Cluster integrity is also protected by evicting or fencing unresponsive nodes.
- Event services, which publishes cluster events so that applications are aware of changes in the cluster
- Time synchronization, which synchronizes the time on all nodes of the cluster
- Network management, which provisions and manages Virtual IP (VIP) addresses that are associated with cluster nodes or application resources to provide a consistent network identity regardless of which nodes are available. In addition, Grid Naming Service (GNS) manages network naming within the cluster.
- High availability, which services, monitors, and restarts all other resources as required
- Cluster Interconnect Link Aggregation (Highly Available IP - HAIP)

Goals for Oracle Clusterware

- Easy installation
- Easy management
- Continuing tight integration with Oracle RAC
- ASM enhancements with benefits for all applications
- No additional clusterware required



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware has become the required clusterware for Oracle Real Application Clusters (RAC). Oracle Database 12c builds on the tight integration between Oracle Clusterware and RAC by extending the integration with Automatic Storage Management (ASM). The result is that now all the shared data in your cluster can be managed by using ASM. This includes the shared data required to run Oracle Clusterware, Oracle RAC, and any other applications you choose to deploy in your cluster.

In most cases, this capability removes the need to deploy additional clusterware from other sources, which also removes the potential for integration issues caused by running multiple clusterware software stacks. It also improves the overall manageability of the cluster.

Although most of the enhancements to ASM are the subject of later lessons, the next part of this lesson examines a series of additional Oracle Clusterware capabilities and the benefits they provide.

Oracle Clusterware Fencing

- An important service provided by Oracle Clusterware is node fencing.
- Node fencing is used to evict nonresponsive hosts from the cluster, preventing data corruptions.
- Oracle Clusterware fencing traditionally used “fast” reboots to enforce node removal.
- Oracle Clusterware now supports rebootless node fencing.
 - Processes performing I/O are identified and terminated on the offending node.
 - Clusterware is then stopped and restarted on that node.
- Oracle Clusterware also supports a fencing mechanism based on remote node-termination incorporating IPMI (Intelligent Platform Management Interface).



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

An important service provided by Oracle Clusterware is node fencing. Node fencing is a technique used by clustered environments to evict nonresponsive or malfunctioning hosts from the cluster. Allowing affected nodes to remain in the cluster increases the probability of data corruption due to unsynchronized database writes.

Traditionally, Oracle Clusterware uses a STONITH (Shoot The Other Node In The Head) comparable fencing algorithm to ensure data integrity in cases, in which cluster integrity is endangered and split-brain scenarios need to be prevented. For Oracle Clusterware this means that a local process enforces the removal of one or more nodes from the cluster (fencing). This approach traditionally involved a forced “fast” reboot of the offending node. A fast reboot is a shutdown and restart procedure that does not wait for any I/O to finish or for file systems to synchronize on shutdown. Starting with Oracle Clusterware 11g Release 2 (11.2.0.2), this mechanism has been changed to prevent such a reboot as much as possible by introducing rebootless node fencing.

Now, when a decision is made to evict a node from the cluster, Oracle Clusterware will first attempt to shut down all resources on the machine that was chosen to be the subject of an eviction. Specifically, I/O generating processes are killed and Oracle Clusterware ensures that those processes are completely stopped before continuing.

If all resources can be stopped and all I/O generating processes can be killed, Oracle Clusterware will shut itself down on the respective node, but will attempt to restart after the stack has been stopped.

If, for some reason, not all resources can be stopped or I/O generating processes cannot be stopped completely, Oracle Clusterware will still perform a reboot.

In addition to this traditional fencing approach, Oracle Clusterware now supports a new fencing mechanism based on remote node termination. The concept uses an external mechanism capable of restarting a problem node without cooperation either from Oracle Clusterware or from the operating system running on that node. To provide this capability, Oracle Clusterware supports the Intelligent Platform Management Interface specification (IPMI), a standard management protocol.

To use IPMI and to be able to remotely fence a server in the cluster, the server must be equipped with a Baseboard Management Controller (BMC), which supports IPMI over a local area network (LAN). After this hardware is in place in every server of the cluster, IPMI can be activated either during the installation of the Oracle Grid Infrastructure or after the installation in course of a postinstallation management task by using CRSCTL.

Oracle Clusterware continues to support third-party cluster solutions. For certified solutions, Oracle Clusterware will integrate with the third-party cluster solution in a way that node membership decisions are deferred to the third-party cluster solution. For Oracle RAC environments, it is worth noticing that Oracle Clusterware is mandatory and provides all required functionality. No other third-party solution should therefore be required.

Cluster Time Synchronization

- Time synchronization between cluster nodes is crucial.
- Asynchronous times can make it harder to manage the cluster as a whole.
 - Timestamps are written using the local node time.
 - Log analysis can be impacted severely if the times in a cluster deviate significantly.
- A central time server, accessed by NTP, is typically used to synchronize the server times.
- To make Clusterware independent from failures of external resources, the Oracle Cluster Time Synchronization Service Daemon can be used.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Time synchronization between cluster nodes is crucial. Although a deviating time between the servers in a cluster does not necessarily lead to instability, asynchronous times can make it harder to manage the cluster as a whole. One reason is that timestamps are written using the local node time. Log analysis can be impacted severely if the times in a cluster deviate significantly.

A central time server in the data center, accessed by NTP (Network Time Protocol), is typically used to synchronize the server times to prevent deviating times between the cluster nodes. It is best to avoid sudden time adjustments on individual nodes, which can lead to node evictions when performed too abruptly. To make the Oracle Grid Infrastructure independent from (failures of) external resources, the new Oracle Cluster Time Synchronization Service Daemon (OCTSSD) can be used alternatively to synchronize the time between the servers in one cluster.

The Oracle CTSS daemon is always installed and will always be running, but is configured in accordance to the configuration found on the system. If NTP is installed on the system, CTSS is started in an Observer Mode, not synchronizing the time. Only if NTP is not present on any server of the cluster, CTSS will be activated in active mode, synchronizing the time in the cluster, using one server as the reference server.

Network Resource Management

- Clusterware manages a local network resource that is responsible for monitoring the network on each member node.
- When a network outage is detected, dependent resources (such as VIPs) are informed and failed over to another node, if required.
- Oracle Clusterware maintains one network resource per subnet in the cluster.
- Multiple subnet support is a feature facilitating the consolidation of applications and databases in the grid infrastructure.
- Multiple subnet support enables independent access of applications and databases using different subnets in the cluster.
 - These appear as an independent environment to both the database and application clients.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

More flexibility has been added regarding network management. Technically, the main enhancement is a network resource managed by Oracle Clusterware. As a local cluster resource, it constantly monitors the network on each server. When a network outage is detected, dependent resources, such as VIPs managed by Oracle Clusterware, are informed and failed over to another node, if required.

Oracle Clusterware maintains one network resource per subnet in the cluster. Multiple subnet support is a feature that facilitates the consolidation of applications and databases in the grid infrastructure. Multiple subnet support enables independent access of applications and databases using different subnets in the cluster, which then appears as an independent environment to both the database and application clients.

Oracle Clusterware Operating System Requirements

- Each server must have an operating system that is certified with the Clusterware version being installed.
- Refer to:
 - The certification matrices in the Oracle Grid Infrastructure Installation Guide for your platform
 - Certification matrices located on My Oracle Support:
<http://www.oracle.com/technetwork/database/clustering/tech-generic-unix-new-166583.html>
- When the operating system is installed and working, install Oracle Clusterware to create the cluster.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

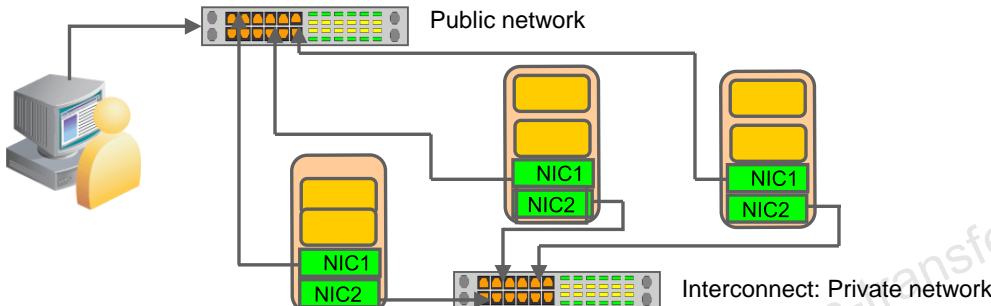
Each server must have an operating system that is certified with the Oracle Clusterware version you are installing. Refer to the certification matrices available in the Oracle Grid Infrastructure Installation Guide for your platform or on My Oracle Support (formerly Oracle MetaLink) for details, which are available from the following URL:

<http://www.oracle.com/technetwork/database/clustering/tech-generic-unix-new-166583.html>

When the operating system is installed and working, you can then install Oracle Clusterware to create the cluster. Oracle Clusterware is installed independently of Oracle Database. After you install Oracle Clusterware, you can then install Oracle Database or Oracle RAC on any of the nodes in the cluster.

Oracle Clusterware Networking

- Each node must have at least two network adapters.
- Each public network adapter must support TCP/IP.
- The interconnect adapter must support:
 - User Datagram Protocol (UDP) or Reliable Data Socket (RDS) for UNIX and Linux for database communication
 - TCP for Windows platforms for database communication
- All platforms use Grid Interprocess Communication (GIPC).



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Each node must have at least two network adapters: one for the public network interface and the other for the private network interface or interconnect. In addition, the interface names associated with the network adapters for each network must be the same on all nodes. For example, in a two-node cluster, you cannot configure network adapters on node1 with eth0 as the public interface, but on node2 have eth1 as the public interface. Public interface names must be the same, so you must configure eth0 as public on both nodes. You should configure the private interfaces on the same network adapters as well. If eth1 is the private interface for node1, eth1 should be the private interface for node2.

Before starting the installation, on each node, you must have at least two interfaces to configure for the public and private IP addresses. You can configure IP addresses with one of the following options:

- Oracle Grid Naming Service (GNS) using one static address defined during installation, which dynamically allocates VIP addresses using Dynamic Host Configuration Protocol (DHCP), which must be running on the network. You must select the Advanced Oracle Clusterware installation option to use GNS.
- Static addresses that network administrators assign on a network domain name server (DNS) or each node. To use the Typical Oracle Clusterware installation option, you must use static addresses.

For the public network, each network adapter must support TCP/IP.

For the private network, the interconnect must support UDP or RDS (TCP for Windows) for communications to the database. Grid Interprocess Communication (GIPC) is used for Grid (Clusterware) interprocess communication. GIPC is a common communications infrastructure to replace CLSC/NS. It provides full control of the communications stack from the operating system up to whatever client library uses it. The dependency on network services (NS) before 11.2 is removed, but there is still backward compatibility with existing CLSC clients (primarily from 11.1). GIPC can support multiple communications types: CLSC, TCP, UDP, IPC, and of course, the communication type GIPC.

Use high-speed network adapters for the interconnects and switches that support TCP/IP. Gigabit Ethernet or an equivalent is recommended.

If you have multiple available network interfaces, Oracle recommends that you use the Redundant Interconnect Usage feature to use multiple interfaces for the private network. However, you can also use third-party technologies to provide redundancy for the private network.

Each node in a cluster requires a supported interconnect protocol to support Cache Fusion and TCP/IP to support Clusterware polling. Token Ring is not supported for cluster interconnects on IBM AIX. Your interconnect protocol must be certified by Oracle for your platform.

Note: Cross-over cables are not supported for use with Oracle Clusterware interconnects.

IP Addresses for Public Networks

- You can configure cluster nodes during installation with either IPv4 or IPv6 addresses on the same network.
- Database clients can connect to either IPv4 or IPv6 addresses.
- The SCAN listener automatically redirects client connection requests to the appropriate database listener for the IP protocol of the client request.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware 12c supports IPv6-based public IP and VIP addresses. IPv6-based IP addresses have become the latest standard for the information technology infrastructure in today's data centers. With this release, Oracle RAC and Oracle Grid Infrastructure support this standard. You can configure cluster nodes during installation with either IPv4 or IPv6 addresses on the same network. Database clients can connect to either IPv4 or IPv6 addresses. The Single Client Access Name (SCAN) listener automatically redirects client connection requests to the appropriate database listener for the IP protocol of the client request.

During installation, Oracle does not support configuring some nodes in the same cluster with all IPv6 addresses and other nodes with all IPv4 addresses, or some nodes with both IPv4 and IPv6 addresses, whereas other nodes are configured exclusively with IPv4 or IPv6 addresses. During installation, you can choose to enable your clients to connect with IPv6 or IPv4 addresses. After installation, you can add IPv4 addresses to an IPv6 cluster or IPv6 addresses to an IPv4 cluster.

Private Network IPv6 Support

- You can now configure cluster nodes to use either IPv4 or IPv6 IP addresses on a private network.
- However, you cannot mix IPv4 and IPv6 addresses for any private network interfaces.
- The IPv6 address format used by Clusterware is defined in RFC 2460.
- The preferred IPv6 address format is as follows:
 - `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`
Each **x** represents a hexadecimal character.
- Empty fields can be collapsed and represented by a '`::`' separator.
 - The address `2001:0db8:0000:0000:0000:8a2e:0370:7334` can be rewritten as `2001:db8::8a2e:370:7334`.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

You can configure cluster nodes to use either IPv4- or IPv6-based IP addresses on a private network, and you can use more than one private network for a cluster. You can configure one or more private network interfaces by using either IPv4 or IPv6 addresses for all the network adapters. However, you cannot mix IPv4 and IPv6 addresses for any private network interfaces. All the nodes in a cluster must use the same IP protocol configuration. All the nodes either use only IPv4 or only IPv6, or all the nodes use both IPv4 and IPv6. You cannot have some nodes in the cluster configured to support only IPv6 addresses, and other nodes in the cluster configured to support only IPv4 addresses.

The preferred IPv6 address format is as follows, where each x represents a hexadecimal character:

`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`

The IPv6 address format is defined by RFC 2460 and Oracle Grid Infrastructure supports IPv6 addresses as follows:

- Global and site-local IPv6 addresses as defined by RFC 4193
- The leading zeros compressed in each field of the IP address
- Empty fields collapsed and represented by a '`::`' separator. For example, you could write the IPv6 address `2001:0db8:0000:0000:0000:8a2e:0370:7334` as `2001:db8::8a2e:370:7334`.

Note: You can only use IPv6 for private networks in clusters using Oracle Clusterware 12c release 2 (12.2) or later.

Grid Naming Service (GNS)

- The only static IP address required for the cluster is the GNS virtual IP address.
- The cluster subdomain is defined as a delegated domain.

```
[root@my-dns-server ~]# cat /etc/named.conf
// Default initial "Caching Only" name server configuration
...
# Delegate to gns on cluster01
cluster01.example.com #cluster sub-domain# NS cluster01-gns.example.com
# Let the world know to go to the GNS vip
cluster01-gns.example.com 192.0.2.155 #cluster GNS Address
```

- A request to resolve cluster01-scan.cluster01.example.com is forwarded to the GNS on 192.0.2.155.
- Each cluster node runs a multicast DNS (mDNS) process.
- You cannot use GNS with another multicast DNS.
 - If you want to use GNS, then disable any third-party mDNS daemons on your system.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Employing Grid Naming Service (GNS) assumes that there is a DHCP server running on the public network with enough addresses to assign to the VIPs and single-client access name (SCAN) VIPs. With GNS, only one static IP address is required for the cluster, the GNS virtual IP address. This address should be defined in the DNS domain. GNS sets up a multicast DNS (mDNS) server within the cluster, which resolves names in the cluster without static configuration of the DNS server for other node IP addresses.

The mDNS server works as follows: Within GNS, node names are resolved using link-local multicast name resolution (LLMNR). It does this by translating the LLMNR “.local” domain used by the multicast resolution to the subdomain specified in the DNS query. When you select GNS, an mDNS server is configured on each host in the cluster. LLMNR relies on the mDNS that Oracle Clusterware manages to resolve names that are being served by that host.

To use GNS before installation, the DNS administrator must establish domain delegation to the subdomain for the cluster. Queries to the cluster are sent to the GNS listener on the GNS virtual IP address. When a request comes to the domain, GNS resolves it using its internal mDNS and responds to the query.

Note: You cannot use GNS with another multicast DNS. If you want to use GNS, then disable any third-party mDNS daemons on your system.

Grid Naming Service Configuration Options

- GNS can run in either automatic or static address configuration mode.
- Automatic configuration occurs in the following ways:
 - For IPv4 addresses, Clusterware assigns identifiers for each node interface, generating names within the delegated subdomain. GNS maintains address and name associations with the addresses leased from the IPv4 DHCP pool.
 - For IPv6 addresses, Clusterware automatically generates addresses with autoconfig.
- With static configurations, no subdomain is delegated.
 - The GNS VIP resolves to an address configured on the DNS.
 - The SCAN resolves to three static addresses for the cluster.
 - Static public and virtual IP names and addresses are defined in the DNS for each cluster member node.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

GNS can run in either automatic or standard cluster address configuration mode. Automatic configuration uses either the DHCP for IPv4 addresses or the Stateless Address Autoconfiguration Protocol (autoconfig) (RFC 2462 and RFC 4862) for IPv6 addresses.

With automatic configurations, a DNS administrator delegates a domain on the DNS to be resolved through the GNS subdomain. During installation, Oracle Universal Installer assigns names for each cluster member node interface designated for Oracle Grid Infrastructure use during installation or configuration. SCANS and all other cluster names and addresses are resolved within the cluster, rather than on the DNS.

Automatic configuration occurs in one of the following ways:

- For IPv4 addresses, Clusterware assigns unique identifiers for each cluster member node interface allocated for Oracle Grid Infrastructure, and generates names using these identifiers within the subdomain delegated to GNS. A DHCP server assigns addresses to these interfaces, and GNS maintains address and name associations with the IPv4 addresses leased from the IPv4 DHCP pool.
- For IPv6 addresses, Clusterware automatically generates addresses with autoconfig.

With static configurations, no subdomain is delegated. A DNS administrator configures the GNS VIP to resolve to a name and address configured on the DNS, and a DNS administrator configures a SCAN name to resolve to three static addresses for the cluster.

A DNS administrator also configures a static public IP name and address, and virtual IP name and address for each cluster member node. A DNS administrator must also configure new public and virtual IP names and addresses for each node added to the cluster. All names and addresses are resolved by DNS.

GNS without subdomain delegation using static VIP addresses and SCANS enables Oracle Flex Cluster and ACFS features that require name resolution information within the cluster. However, any node additions or changes must be carried out as manual administration tasks.

Shared GNS Across Multiple Clusters

- In previous releases, the GNS was dedicated to a single Oracle Grid Infrastructure–based cluster.
- One GNS can now manage just the nodes in its own cluster, or all nodes across all clusters in the data center that are delegated to GNS for resolution.
- Using only one GNS for all nodes that are part of a Grid Infrastructure cluster in the data center:
 - Streamlines the naming convention
 - Enables a data center cloud
 - Minimizes day-to-day administration efforts



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Highly Available Grid Naming Service

- Shared GNS High Availability provides high availability of lookup and other services to the clients.
- Highly available GNS consists of one primary GNS instance and zero or more secondary GNS instances.
 - The primary GNS instance services all updates from clients.
 - Both the primary and secondary GNS process look up queries.
- The secondary GNS acts as backup for the primary GNS.
- Secondary GNS instances can be promoted to the primary role whenever an existing primary GNS fails.
- The primary GNS manages zone data and holds all records on the delegated domain.
- Zone data and its change history are stored in the OCR.
- Secondary GNS zone data updates require a zone transfer.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Highly available GNS consists of one primary GNS instance and zero or more secondary GNS instances. The primary GNS instance services all updates from clients, whereas both the primary and the secondary GNS instances process the lookup queries. Additionally, the secondary GNS instances act as backup for the primary GNS instance. Secondary GNS instances can be promoted to the primary role whenever an existing primary GNS instance fails or is removed by the cluster administrator.

Highly available GNS provides fault tolerance by taking data backup on a secondary GNS instance by using zone transfer. Secondary GNS instances get a copy of data from the primary GNS instance during installation. Thereafter, any update on the primary GNS instance gets replicated to the secondary GNS instances.

The primary GNS instance manages zone data and holds all records on the delegated domain. It stores the zone data and its change history in the Oracle Cluster Registry (OCR). Updating zone data on a secondary GNS instance involves a zone transfer, which can be one of two methods:

- **Full zone transfer:** The primary GNS instance replicates all zone data to the secondary GNS instances.

- **Incremental zone transfer:** The primary GNS instance replicates only the changed data to the secondary GNS instances. GNS uses this transfer mechanism for the following scenarios:
 - When there is an update to the zone data in the primary GNS instance, the instance notifies the secondary instances to initiate a data transfer. The secondary GNS instances will ask for a data transfer only if the serial number of the data in the OCR of the primary GNS instance is greater than that of the data of the secondary GNS instances.
 - When the refresh time of a secondary GNS instance expires, the instance sends a query containing its data serial number to the primary GNS instance. If the serial number of the secondary GNS instance is less than that of the primary GNS instance, GNS initiates a zone transfer.

You must configure a primary GNS instance before you configure any secondary instance. After you successfully configure a primary GNS instance, you export client data for clients and secondary GNS instances. You provide exported client data when you configure secondary GNS instances. All secondary GNS instances register themselves with the primary GNS instance and get a copy of the zone data. Secondary GNS instances contact the primary GNS instance for data updates by using the zone transfer mechanism, when either the refresh time of the secondary GNS instance expires or in response to a notification.

Configuring Highly Available GNS

1. As the cluster administrator, configure and start the primary GNS instance on any node in an existing cluster.

```
# srvctl add gns -vip gns_vip -domain gns_subdomain  
# srvctl start gns
```

2. Create client data for the secondary GNS instances.

```
# srvctl export gns -clientdata file_name -role secondary
```

3. Copy the client data file that you created in the preceding step to the secondary GNS instance.

4. Configure and start the secondary GNS instance.

```
# srvctl add gns -vip gns_vip -clientdata file_name  
# srvctl start gns
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Single-Client Access Name

- The single-client access name (SCAN) is the address used by clients connecting to the cluster.
- The SCAN is a fully qualified host name located in the GNS subdomain registered to three IP addresses.

```
$ nslookup cluster01-scan.cluster01.example.com
Server:          192.0.2.1
Address:         192.0.2.1#53

Non-authoritative answer:
Name:   cluster01-scan.cluster01.example.com
Address: 192.0.2.243
Name:   cluster01-scan.cluster01.example.com
Address: 192.0.2.244
Name:   cluster01-scan.cluster01.example.com
Address: 192.0.2.245
```

- The SCAN provides a stable, highly available name for clients to use, independent of the nodes that make up the cluster.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The single-client access name (SCAN) is the address used by clients connecting to the cluster. The SCAN is a fully qualified host name (host name + domain) registered to three IP addresses. If you use GNS, and have DHCP support, then the GNS will assign addresses dynamically to the SCAN.

If you do not use GNS, the SCAN should be defined in the DNS to resolve to the three addresses assigned to that name. This should be done before you install Oracle Grid Infrastructure. The SCAN and its associated IP addresses provide a stable name for clients to use for connections, independent of the nodes that make up the cluster.

SCANS function like a cluster alias. However, SCANS are resolved on any node in the cluster, so unlike a VIP address for a node, clients connecting to the SCAN no longer require updated VIP addresses as nodes are added to or removed from the cluster. Because the SCAN addresses resolve to the cluster, rather than to a node address in the cluster, nodes can be added to or removed from the cluster without affecting the SCAN address configuration.

During installation, listeners are created on each node for the SCAN IP addresses. Oracle Clusterware routes application requests to the cluster SCAN to the least loaded instance providing the service.

SCAN listeners can run on any node in the cluster. SCANS provide location independence for databases so that the client configuration does not have to depend on which nodes run a particular database.

Instances register with SCAN listeners only as remote listeners. Upgraded databases register with SCAN listeners as remote listeners, and also continue to register with all other listeners.

If you specify a GNS domain during installation, the SCAN defaults to *clusternname-scan.GNS_domain*. If a GNS domain is not specified at installation, the SCAN defaults to *clusternname-scan.current_domain*.



Quiz

Which of the following statements about Grid Naming Service is *not* true?

- a. GNS is an integral component of Grid Plug and Play.
- b. Each node in the cluster runs a multicast DNS (mDNS) process.
- c. The GNS virtual IP address must be assigned by DHCP.
- d. The cluster subdomain is defined as a delegated domain.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.



Quiz

Each cluster node's public Ethernet adapter must support UDP or RDS.

- a. True
- b. False



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Explain the principles and purposes of clusters
- Describe cluster hardware best practices
- Describe the Oracle Clusterware architecture



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware Architecture



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to describe Oracle Clusterware:

- Architecture
- Startup details



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware Technology Stack

- Clusterware is a platform-independent facility for starting, stopping, and managing clusterwide resources.
- Oracle Clusterware comprises two physical stacks:
 - Cluster Ready Services technology stack
 - Oracle High Availability Services (OHAS) technology stack



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware consists of two separate stacks: an upper stack anchored by the Cluster Ready Services (CRS) daemon (`crsd`) and a lower stack anchored by the Oracle High Availability Services daemon (`ohasd`). These two stacks have several processes that facilitate cluster operations.

The Cluster Ready Services stack manages cluster resources based on the configuration information that is stored in OCR for each resource. This includes start, stop, monitor, and failover operations.

The Oracle High Availability Services (OHAS) stack is responsible for monitoring and maintaining the high availability of Oracle ASM and Oracle Clusterware itself.

Cluster Ready Services Technology Stack

Components of the Cluster Ready Services technology stack:

- Cluster Ready Services
- Cluster Synchronization Services
- Oracle ASM
- Cluster Time Synchronization Service
- Event Management
- Grid Naming Service
- Oracle Agent
- Oracle Notification Service
- Oracle Root Agent



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

- **Cluster Ready Services (CRS):** The primary program for managing high availability operations in a cluster
The CRSD manages cluster resources based on the configuration information that is stored in OCR for each resource. This includes start, stop, monitor, and failover operations. The CRSD process generates events when the status of a resource changes. When you have Oracle RAC installed, the CRSD process monitors the Oracle database instance, listener, and so on, and automatically restarts these components when a failure occurs.
- **Cluster Synchronization Services (CSS):** Manages the cluster configuration by controlling which nodes are members of the cluster and by notifying members when a node joins or leaves the cluster. If you are using certified third-party clusterware, CSS processes interface with your clusterware to manage node membership information. The cssdagent process monitors the cluster and provides I/O fencing. This service formerly was provided by Oracle Process Monitor Daemon (`oprocd`), also known as `OraFenceService` on Windows. A cssdagent failure may result in Oracle Clusterware restarting the node.
- **Oracle ASM:** Provides disk management for Oracle Clusterware and Oracle Database
- **Cluster Time Synchronization Service (CTSS):** Provides time management in a cluster for Oracle Clusterware

- **Event Management (EVM):** A background process that publishes events that Oracle Clusterware creates
- **Grid Naming Service (GNS):** Handles requests sent by external DNS servers, performing name resolution for names defined by the cluster
- **Oracle Agent (oraagent):** Extends clusterware to support Oracle-specific requirements and complex resources. This process runs server callout scripts when FAN events occur. This process was known as RACG in Oracle Clusterware 11g release 1 (11.1).
- **Oracle Notification Service (ONS):** A publish-and-subscribe service for communicating Fast Application Notification (FAN) events
- **Oracle Root Agent (orarootagent):** A specialized oraagent process that helps the CRSD manage resources owned by root, such as the network, and the Grid virtual IP address

The Cluster Synchronization Service (CSS), Event Management (EVM), and Oracle Notification Services (ONS) components communicate with other cluster component layers on other nodes in the same cluster database environment. These components are also the main communication links between Oracle Database, applications, and the Oracle Clusterware high availability components. In addition, these background processes monitor and manage database operations.

OHAS Technology Stack

Components of the OHAS technology stack:

- Appagent
- Cluster Logger Service
- Grid Interprocess Communication
- Grid Plug and Play
- Multicast Domain Name Service
- Oracle Agent
- Oracle Root Agent
- Scriptagent
- System Monitor Service



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

- **Appagent:** Protects any resources of the application resource type used in previous versions of Oracle Clusterware
- **Cluster Logger Service (ologgerd):** Receives information from all the nodes in the cluster and persists in a Oracle Grid Infrastructure Management Repository-based database. This service runs on only two nodes in a cluster. The cluster logger service manages the operating system metric database in the Grid Infrastructure Management Repository.
- **Grid Interprocess Communication (GIPC):** A support daemon that enables Redundant Interconnect Usage
- **Grid Plug and Play (GPNP):** Provides access to the Grid Plug and Play profile, and coordinates updates to the profile among the nodes of the cluster to ensure that all of the nodes have the most recent profile
- **Multicast Domain Name Service (mDNS):** Used by Grid Plug and Play to locate profiles in the cluster, as well as by GNS to perform name resolution. The mDNS process is a background process on Linux, UNIX, and Windows.
- **Oracle Agent (oraagent):** Extends clusterware to support Oracle-specific requirements and complex resources. This process manages daemons that run as the Oracle Clusterware owner, such as the GIPC, GPNPD, and GIPC daemons.

- **Oracle Root Agent (orarootagent):** A specialized oraagent process that helps the CRSD manage resources owned by root, such as the Cluster Health Monitor (CHM)
- **Scriptagent:** Protects resources of resource types other than application when using shell or batch scripts to protect an application
- **System Monitor Service (osysmond):** The monitoring and operating system metric collection service that sends the data to the cluster logger service. This service runs on every node in a cluster.

Clusterware Component Processes and Services

| Component | Linux/UNIX Process | Windows Processes |
|---------------------------|---|--|
| CRS | crsd.bin (r) | crsd.exe |
| CSS | ocssd.bin, cssdmonitor(r), cssdagent(r) | cssdagent.exe, cssdmonitor.exe ocssd.exe |
| CTSS | octssd.bin (r) | octssd.exe |
| EVM | evmd.bin, evmlogger.bin | evmd.exe |
| GIPC | gipcd.bin | N/A |
| GNS | gnsd (r) | gnsd.exe |
| Grid Plug and Play | gpnpd.bin | gpnpd.exe |
| Logger | ologgerd.bin (r) | ologgerd.exe |
| Master Diskmon | diskmon.bin | N/A |
| mDNS | mdnsd.bin | mDNSResponder.exe |
| Oracle Agent | oraagent.bin | oraagent.exe |
| OHAS | ohasd.bin (r) | ohasd.exe |
| ONS | ons | ons.exe |
| Oracle Root Agent | Orarootagent.bin (r) | orarootagent.exe |
| Sysmon | osysmond.bin (r) | osysmond.exe |



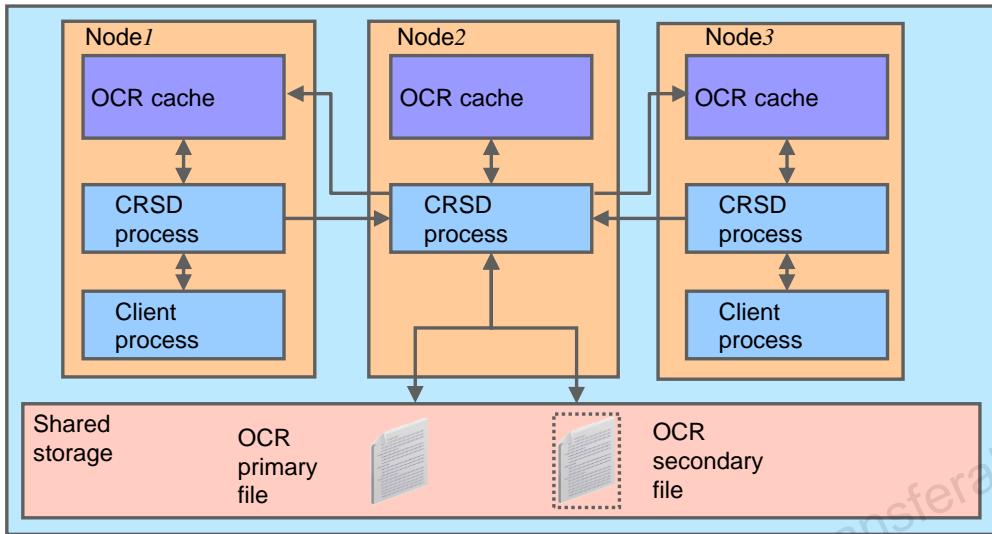
Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The table in the slide lists the processes and services associated with Oracle Clusterware components. Note that Oracle ASM is not just one process, but an instance. Given Oracle Flex ASM, Oracle ASM does not necessarily run on every cluster node but only on some of them.

The only Windows services associated with the Oracle Grid Infrastructure are OracleOHService (OHASD), Oracle ASM, listener services (including node listeners and SCAN listeners), and management database. Oracle ASM can be considered part of the Oracle Clusterware technology stack when OCR is stored on Oracle ASM. The listeners and management database are Oracle Clusterware resources and are not properly part of the Oracle Clusterware technology stack.

Note: In the table in the slide, if a UNIX or a Linux system process has an (r) beside it, the process runs with root privileges.

Oracle Clusterware Repository (OCR)



ORACLE

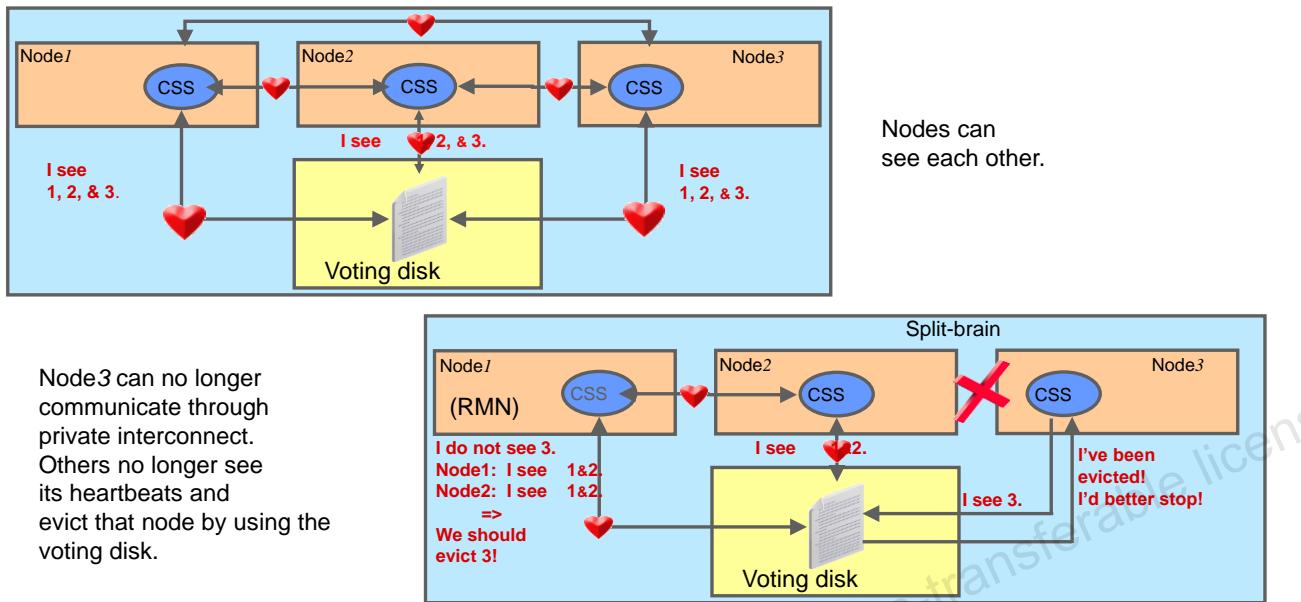
Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Cluster configuration information is maintained in the OCR. You can have up to five OCR locations. Each OCR location must reside on shared storage that is accessible by all of the hub nodes in the cluster. The OCR relies on distributed shared cache architecture for optimizing queries, and clusterwide atomic updates against the cluster repository. Each node in the cluster maintains an in-memory copy of OCR, along with the CRSD that accesses its OCR cache. Only one of the CRSD processes actually reads from and writes to the OCR file on shared storage. This process is responsible for refreshing its own local cache, as well as the OCR cache on other nodes in the cluster. For queries against the cluster repository, the OCR clients communicate directly with the local CRS daemon (CRSD) process on the node from which they originate. When clients need to update the OCR, they communicate through their local CRSD process to the CRSD process that is performing input/output (I/O) for writing to the repository on disk.

The main OCR client applications are OUI, SRVCTL, Enterprise Manager (EM), the Database Configuration Assistant (DBCA), the Database Upgrade Assistant (DBUA), Network Configuration Assistant (NETCA), and the ASM Configuration Assistant (ASMCA). Furthermore, OCR maintains dependency and status information for application resources defined within Oracle Clusterware, specifically databases, instances, services, and node applications.

Note: In the diagram in the slide, note that a client process might also exist on node 2 but is not shown for the sake of clarity.

CSS Voting Disk Function



ORACLE®

CSS is the service that determines which nodes in the cluster are available and provides cluster group membership and simple locking services to other processes. CSS typically determines node availability via communication through a dedicated private network with a voting disk used as a secondary communication mechanism. This is done by sending heartbeat messages through the network and the voting disk, as illustrated by the top graphic in the slide. The voting disk is a file on a clustered file system that is accessible to all nodes in the cluster. Its primary purpose is to help in situations where the private network communication fails. The voting disk is then used to communicate the node state information used to determine which nodes go offline. Without the voting disk, it can be difficult for isolated nodes to determine whether it is experiencing a network failure or whether the other nodes are no longer available. It would then be possible for the cluster to enter a state where multiple subclusters of nodes would have unsynchronized access to the same database files. The bottom graphic illustrates what happens when Node3 can no longer send heartbeats to other members of the cluster. When others can no longer see Node3's heartbeats, they decide to evict that node by using the voting disk. When Node3 reads the removal message or "kill block," it generally reboots itself to ensure that all outstanding write I/Os are lost.

Voting Disk Considerations

- If you configure voting disks on Oracle ASM:
 - You do not need to manually configure the voting disks
 - An appropriate number of voting disks will be created depending on the disk group redundancy
- Desupport of Direct File System Placement for Oracle Cluster Registry (OCR) and Voting Files:
 - Starting with Oracle Grid Infrastructure 12c Release 2 (12.2), the placement of Oracle Clusterware files: the Oracle Cluster Registry (OCR), and the Voting Files, directly on a shared file system is desupported in favor of having Oracle Clusterware files managed by Oracle Automatic Storage Management (Oracle ASM).



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

If you configure voting disks on Oracle ASM, then you do not need to manually configure the voting disks. Depending on the redundancy of your disk group, an appropriate number of voting disks are created.

Starting with Oracle Grid Infrastructure 12c Release 2 (12.2), the placement of Oracle Clusterware files: the Oracle Cluster Registry (OCR), and the Voting Files, directly on a shared file system is desupported in favor of having Oracle Clusterware files managed by Oracle Automatic Storage Management (Oracle ASM). You cannot place Oracle Clusterware files directly on a shared file system. If you need to use a supported shared file system, either a Network File System, or a shared cluster file system instead of native disk devices, then you must create Oracle ASM disks on supported network file systems that you plan to use for hosting Oracle Clusterware files before installing Oracle Grid Infrastructure. You can then use the Oracle ASM disks in an Oracle ASM disk group to manage Oracle Clusterware files.

If your Oracle Database files are stored on a shared file system, then you can continue to use shared file system storage for database files, instead of moving them to Oracle ASM storage.

Oracle Local Registry and High Availability

- The Oracle Local Registry (OLR) is a registry similar to OCR that is located on each node in a cluster.
 - The OLR contains Clusterware manageability information, including dependencies between services.
- Oracle High Availability Services uses the OLR to maintain Clusterware resources.
- OLR is located on local storage on each node in a cluster.
- The Default location is:
Grid_home/cdata/host_name.olr



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

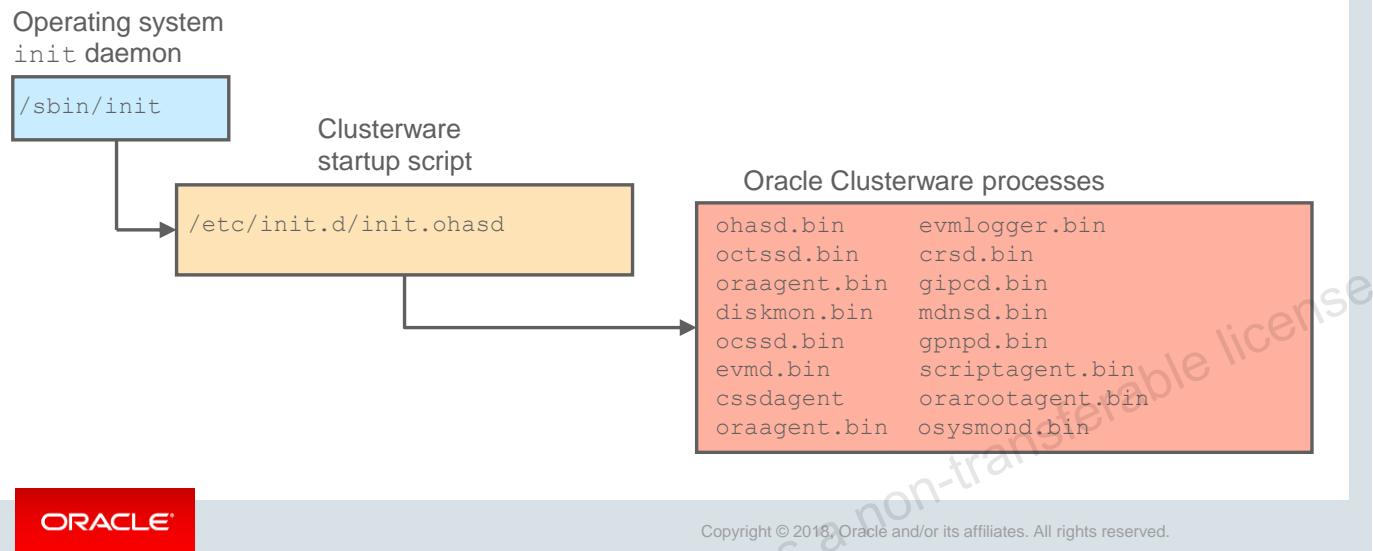
In Oracle Clusterware 12c, each node in a cluster has a local registry for node-specific resources, called an Oracle Local Registry (OLR). The OLR is a registry similar to the OCR, but contains information specific to each node. It contains manageability information about Oracle Clusterware, including dependencies between various services. Oracle High Availability Services uses this information. 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.

OLR is located on local storage on each node in a cluster. Its default location is in the path *Grid_home/cdata/host_name.olr*, where *Grid_home* is the Oracle Grid Infrastructure home, and *host_name* is the host name of the node.

The OLR is backed up at the end of an installation or an upgrade. After that time, you can only manually back up the OLR. Automatic backups are not supported for the OLR. You should create a new backup when you migrate OCR from Oracle ASM to other storage, or when you migrate OCR from other storage to Oracle ASM.

Oracle Clusterware Initialization

The init process calls the `/etc/init.d/init.ohasd` script, which starts critical Clusterware processes.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

During the installation of Oracle Clusterware, the `init.ohasd` startup script is copied to `/etc/init.d`. The wrapper script is responsible for setting up environment variables, and then starting the Oracle Clusterware daemons and processes.

The Oracle High Availability Services daemon (`ohasd`) is responsible for starting in proper order, monitoring, and restarting other local Oracle daemons, including the `crsd` daemon, which manages clusterwide resources. When `init` starts `ohasd` on Clusterware startup, `ohasd` starts `orarootagent`, `cssdagent`, and `oraagent`. Some of the high-availability daemons will be running under the `root` user with real-time priority, and others will be running under the Clusterware owner with user-mode priorities after they are started. When a command is used to stop Oracle Clusterware, the daemons will be stopped, but the `ohasd` process will remain running.

Oracle Clusterware Initialization

- Oracle Clusterware is started by the OS init daemon calling the /etc/init.d/init.ohasd startup script.
- On OL6, Clusterware startup is controlled by Upstart via the /etc/init/oracle-ohasd.conf file.

```
# cat /etc/init/oracle-ohasd.conf
# Oracle OHASD startup

start on runlevel [35]
stop  on runlevel [!35]
respawn
exec /etc/init.d/init.ohasd run >/dev/null 2>&1 </dev/null
```

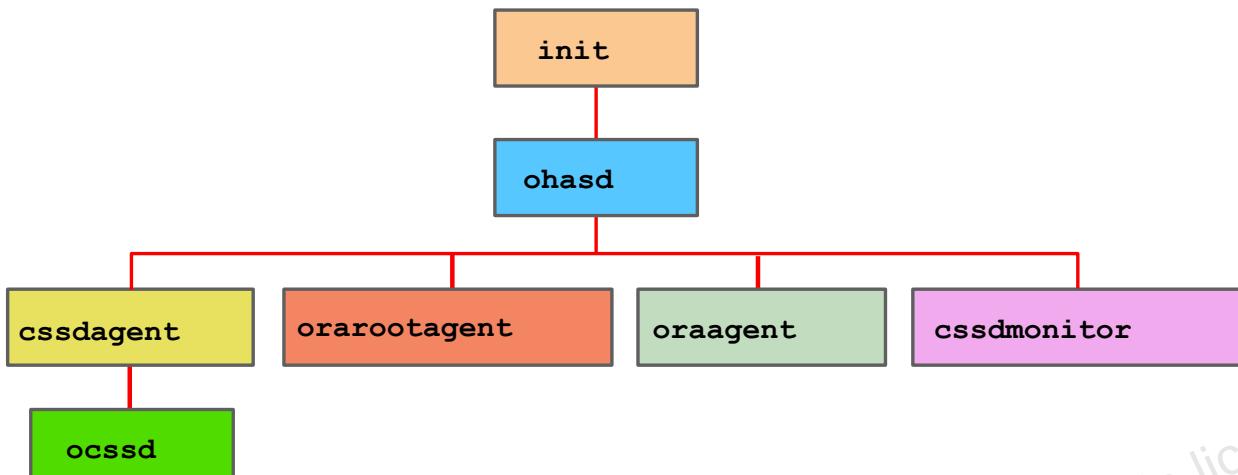
- On OL7, Clusterware startup is controlled by systemd to manage start/stop services (example: /etc/systemd/system/oracle-ohasd.service)



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Linux 6 (OL6) or Red Hat Linux 6 (RHEL6) has deprecated inittab, rather, init.ohasd will be configured via upstart in /etc/init/oracle-ohasd.conf, however, the process /etc/init.d/init.ohasd run should still be up. Oracle Linux 7 (and Red Hat Linux 7) uses systemd to manage start/stop services (example: /etc/systemd/system/oracle-ohasd.service)

Clusterware Startup Details



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

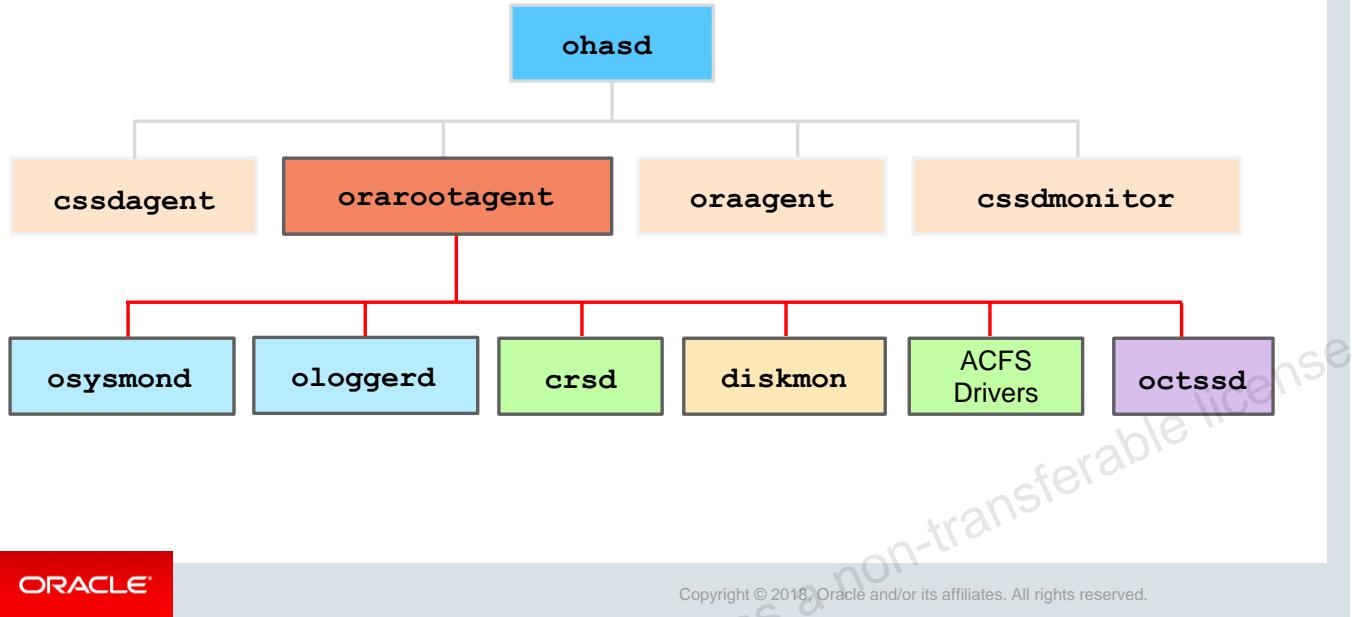
The Oracle High Availability Services daemon (`ohasd`) is responsible for starting in proper order, monitoring, and restarting other local Oracle Clusterware daemons, up through the `crsd` daemon, which in turn manages clusterwide resources.

When a cluster node boots, or Clusterware is started on a running clusterware node, the `init` process starts `ohasd`. The `ohasd` process then initiates the startup of the processes in the lower, or Oracle High Availability (OHASD) stack.

- The `cssdagent` process is started, which in turn starts `ocssd`. The `ocssd` process discovers the voting disk either in ASM or on shared storage, and then joins the cluster. The `cssdagent` process monitors the cluster and provides I/O fencing. This service formerly was provided by Oracle Process Monitor Daemon (`oprocd`). A `cssdagent` failure may result in Oracle Clusterware restarting the node.
- The `orarootagent` is started. This process is a specialized `oraagent` process that helps `crsd` start and manage resources owned by root, such as the network and the grid virtual IP address.

- The `oraagent` process is started. It is responsible for starting processes that do not need to be run as `root`. The `oraagent` process extends Clusterware to support Oracle-specific requirements and complex resources. This process runs server-callout scripts when FAN events occur. This process was known as RACG in Oracle Clusterware 11g Release 1 (11.1).
- The `cssdmonitor` is started and is responsible for monitoring the `ocssd` daemon.

Clusterware Startup: OHASD orarootagent



ORACLE®

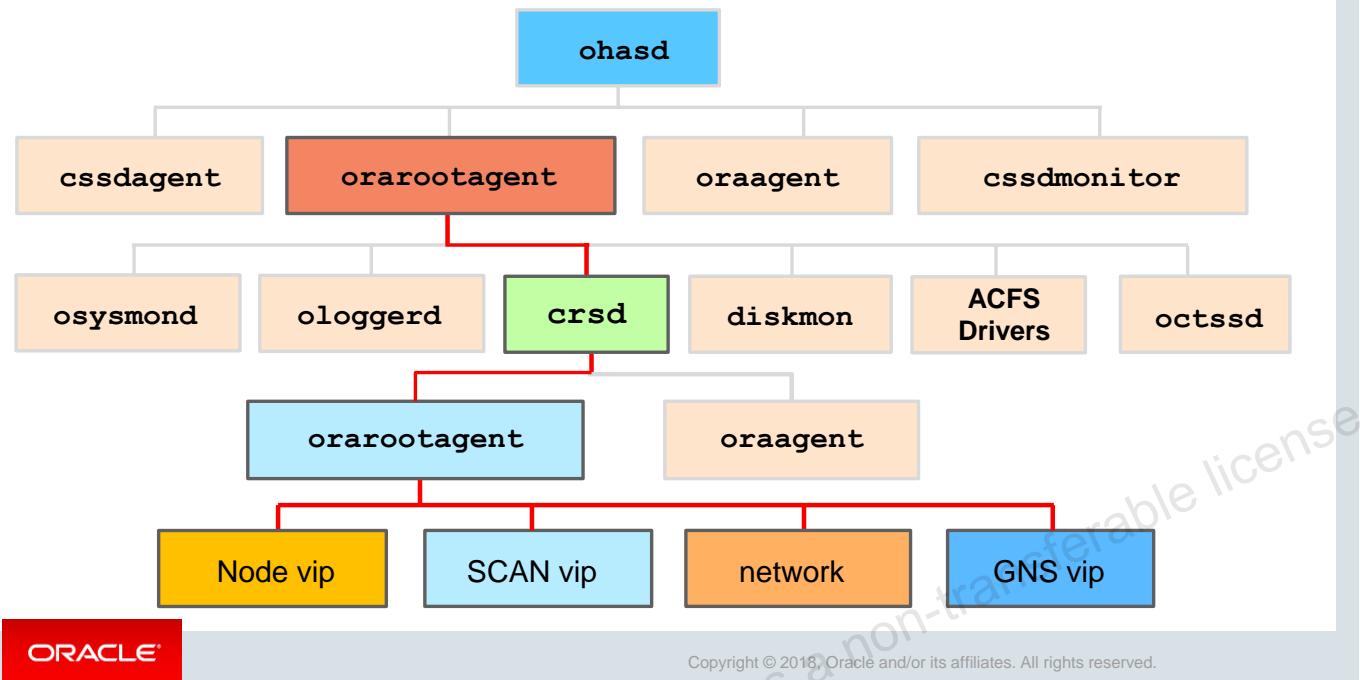
Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The **orarootagent** process is responsible for starting the following processes:

- **osysmond**: The system monitor service (**osysmond**) is the monitoring and operating system metric collection service that sends data to the cluster logger service, **ologgerd**. The cluster logger service receives information from all the nodes and persists in an Oracle Grid Infrastructure Management Repository database. There is one system monitor service on every node.
- **ologgerd**: There is a cluster logger service (**ologgerd**) on only one node in a cluster and another node is chosen by the cluster logger service to house the standby for the master cluster logger service. If the master cluster logger service fails, the node where the standby resides takes over as master and selects a new node for standby. The master manages the operating system metric database in the CHM repository and interacts with the standby to manage a replica of the master operating system metrics database.
- **crsd**: The Cluster Ready Services (CRS) process is the primary program for managing high availability operations in a cluster. The CRS daemon (**crsd**) manages cluster resources based on the configuration information stored in OCR for each resource. This includes start, stop, monitor, and failover operations. The **crsd** process generates events when the status of a resource changes. When Oracle RAC is installed, the **crsd** process monitors the Oracle database components and automatically restarts them when a failure occurs.

- **diskmon:** The diskmon process monitors and performs I/O fencing for Oracle Exadata.
- **ACFS Drivers:** These drivers are loaded in support of ASM Dynamic Volume Manager (ADVM) and ASM Cluster File System (ACFS).
- **ctssd:** The Cluster Time Synchronization Service process provides time synchronization for the cluster in the absence of ntpd. If ntpd is configured, octssd will run in observer mode.

Clusterware Startup Details: CRSD orarootagent



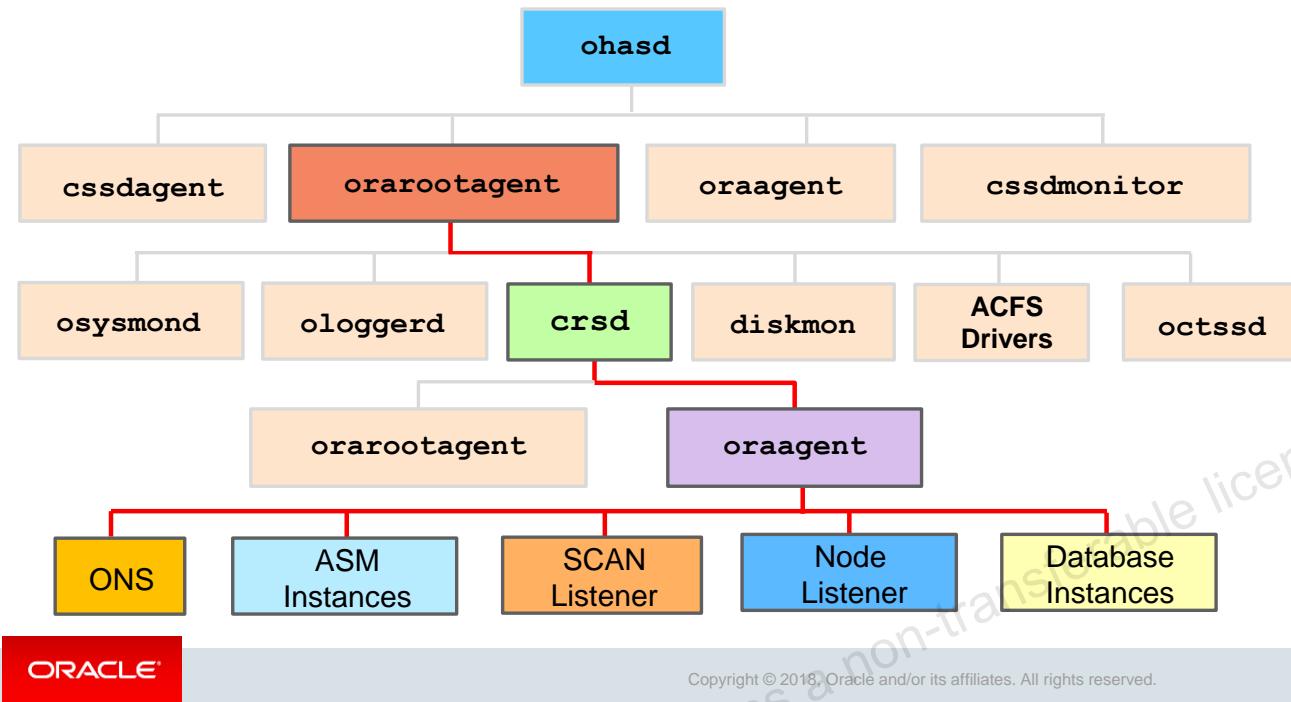
ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

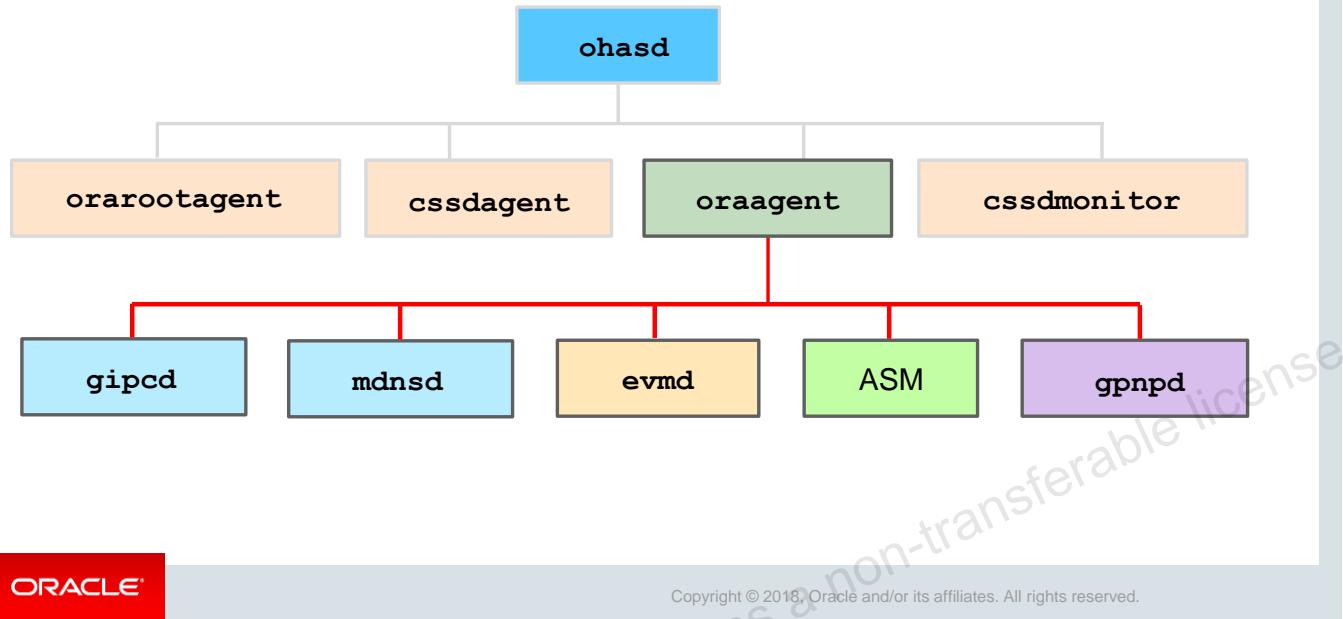
The `crsd` process starts another `orarootagent` process and another `oraagent` process. The new `orarootagent` process is responsible for starting the following resources:

- **Node vip:** The node vip is a node application (nodeapp) responsible for eliminating response delays (TCP timeouts) to client programs requesting a connection to the database. Each node vip is assigned an unused IP address. This is usually done via DHCP but can be manually assigned. There is initially one node vip per cluster node at Clusterware startup. When a cluster node becomes unreachable, the node vip is failed over to a surviving node and redirects connection requests made to the unreachable node to a surviving node.
- **SCAN vip:** SCAN vips or Single Client Access Name vips are part of a connection framework that eliminates dependencies on static cluster node names. This framework allows nodes to be added to or removed from the cluster without affecting the ability of clients to connect to the database. If GNS is used in the cluster, three SCAN vips are started on the member nodes using the IP addresses assigned by the DHCP server. If GNS is not used, SCAN vip addresses for the cluster can be defined in the DNS server used by the cluster nodes.
- **Network:** Network resources required by the cluster are started.
- **GNS vip:** If GNS is used to resolve client requests for the cluster, a single GNS vip for the cluster is started. The IP address is assigned in the GNS server used by the cluster nodes.

Clusterware Startup Details: CRSD oraagent



Clusterware Startup Details: OHASD oraagent



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The **oraagent** process started by **ohasd** is responsible for starting the following processes:

- **gipcd**: The Grid Interprocess Communication (GIPC) daemon is a support process that enables Redundant Interconnect Usage. Redundant Interconnect Usage enables load-balancing and high availability across multiple (up to four) private networks (also known as interconnects).
- **mdnsd**: The Multicast Domain Name Service (mDNS) daemon is used by Grid Plug and Play to locate profiles in the cluster, as well as by GNS to perform name resolution.
- **evmd**: The Event Management (EVM) daemon is a background process that publishes the events that Oracle Clusterware creates.
- **ASM**: ASM provides disk management for Oracle Clusterware and Oracle Database.
- **gpnpd**: The Grid Plug and Play daemon (GPNPD) provides access to the Grid Plug and Play profile and coordinates updates to the profile among the nodes of the cluster to ensure that all the nodes have the most recent profile.

Controlling Oracle Clusterware

The `crsctl` utility is used to invoke certain OHASD functions.

- To stop or start Oracle Clusterware on the local node:

```
# crsctl stop cluster  
# crsctl start cluster
```

- To start Oracle Clusterware on all, all hub, or all leaf nodes:

```
# crsctl start cluster -all | -hub | -leaf
```

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

```
# crsctl enable crs  
# crsctl disable crs
```

- To check the status of CRS on the local node:

```
# crsctl check cluster
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

When a node that contains Oracle Grid Infrastructure is started, `ohasd` is automatically started by `init`. When the `crsctl` utility is used to disable Cluster Ready Services (CRS) from automatically starting, state information that is related to startup is placed in the `SCLS_SRC` control files, preventing automatic startup on machine reboot.

Looking at the `crsstart` file on a normally running cluster node shows the following:

```
# cat /etc/oracle/scls_scr/host01/root/ohasdstr  
enable
```

Executing the `crsctl disable crs` command yields the following output:

```
# crsctl disable crs  
CRS-4621: Oracle High Availability Services autostart is disabled.
```

Looking again at the `crsstart` file now shows this:

```
# cat /etc/oracle/scls_scr/host01/root/ohasdstr  
disable
```

When the node is rebooted, OHAS services will not be started on that node.

To check the status of CRS on all nodes, use the following syntax:

```
# crsctl check cluster -all
```

Verifying the Status of Oracle Clusterware

The `crsctl` utility can be used to verify the status of Oracle Clusterware on all nodes:

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



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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, the `check` commands do not need to be run by the `root` user 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. This command is 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.

Viewing the High Availability Services Stack

```
$ crsctl stat res -init -t
```

| NAME | TARGET | STATE | SERVER | STATE_DETAILS |
|-------------------------------|--------|--------|--------|------------------------|
| Cluster Resources | | | | |
| ora.asm | 1 | ONLINE | ONLINE | host01 Started |
| ora.cluster_interconnect.haip | 1 | ONLINE | ONLINE | host01 STABLE |
| ora.crf | 1 | ONLINE | ONLINE | host01 STABLE |
| ora.crsd | 1 | ONLINE | ONLINE | host01 STABLE |
| ora.cssd | 1 | ONLINE | ONLINE | host01 STABLE |
| ora.cssdmonitor | 1 | ONLINE | ONLINE | host01 STABLE |
| ora.ctssd | 1 | ONLINE | ONLINE | host01 OBSERVER,STABLE |
| ora.evmd | 1 | ONLINE | ONLINE | host01 STABLE |
| ... | | | | |

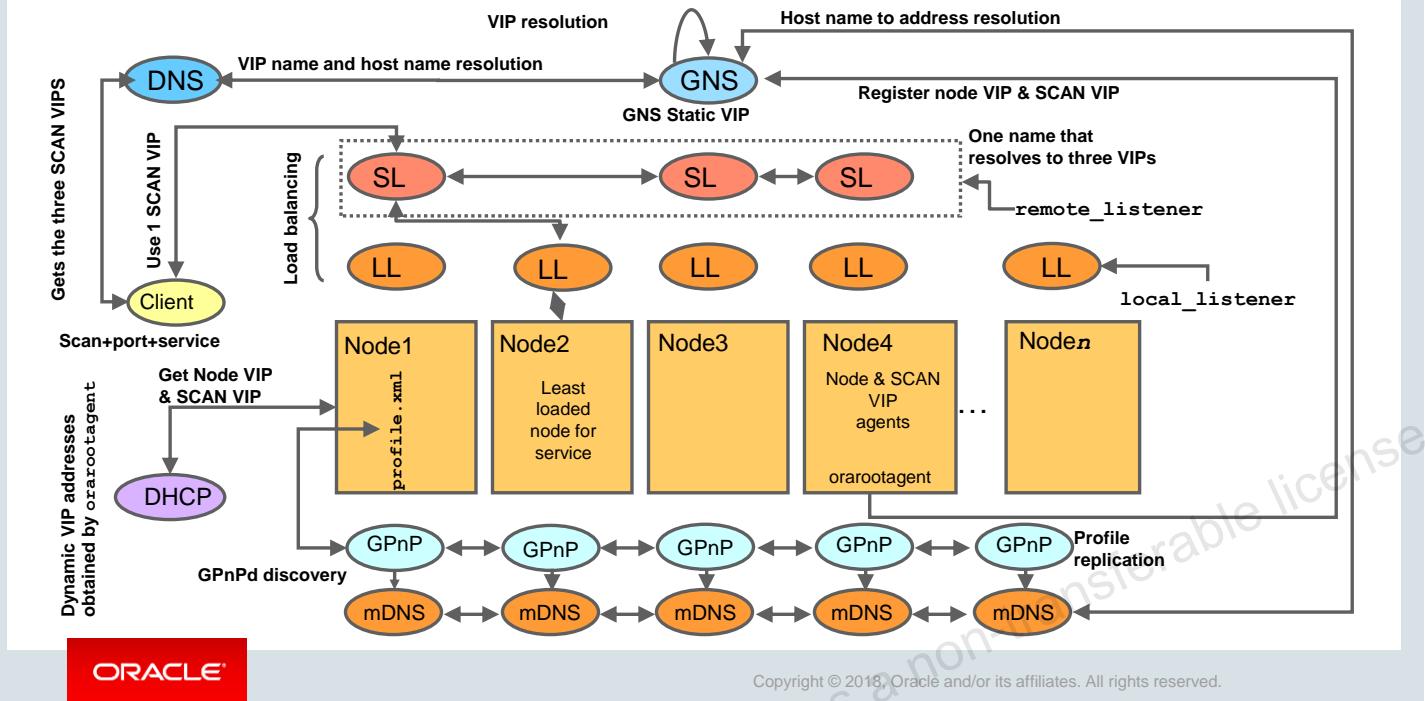


Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

To view the Oracle High Availability stack, use the `crsctl` command as follows:

```
$ crsctl stat res -init -t
```

GnP Architecture: Overview



GPnP Service

The GPnP service is collectively provided by all the GPnP agents. It is a distributed method of replicating profiles. The service is instantiated on each node in the domain as a GPnP agent. The service is peer-to-peer; there is no master process. This allows high availability because any GPnP agent can crash and new nodes will still be serviced. GPnP requires standard IP multicast protocol (provided by mDNS) to locate peer services. Using multicast discovery, GPnP locates peers without configuration. This is how a GPnP agent on a new node locates another agent that may have a profile it should use.

Name Resolution

A name defined within a GPnP domain is resolvable in the following cases:

- Hosts inside the GPnP domain use normal DNS to resolve the names of hosts outside of the GPnP domain. They contact the regular DNS service and proceed. They may get the address of the DNS server by global configuration or by having been told by DHCP.
 - Within the GPnP domain, host names are resolved using mDNS. This requires an mDNS responder on each node that knows the names and addresses used by this node, and operating system client library support for name resolution using this multicast protocol. Given a name, a client executes `gethostbyname`, resulting in an mDNS query. If the name exists, the responder on the node that owns the name will respond with the IP address.

The client software may cache the resolution for the given time-to-live value.

- Machines outside the GPnP domain cannot resolve names in the GPnP domain by using multicast. To resolve these names, they use their regular DNS. The provisioning authority arranges the global DNS to delegate a subdomain (zone) to a known address that is in the GPnP domain. GPnP creates a service called GNS to resolve the GPnP names on that fixed address.

The node on which the GNS server is running listens for DNS requests. On receipt, they translate and forward to mDNS, collect responses, translate, and send back to the outside client. GNS is “virtual” because it is stateless. Any node in the multicast domain may host the server.

The only GNS configuration is global:

- The address on which to listen on standard DNS port 53
- The name(s) of the domains to be serviced

There may be as many GNS entities as needed for availability reasons. Oracle-provided GNS may use CRS to ensure availability of a single GNS provider.

SCAN and Local Listeners

When a client submits a connection request, the SCAN listener listening on a SCAN IP address and the SCAN port are contacted on the client’s behalf. Because all services on the cluster are registered with the SCAN listener, the SCAN listener replies with the address of the local listener on the least-loaded node where the service is currently being offered. Finally, the client establishes a connection to the service through the listener on the node where service is offered. All these actions take place transparently to the client without any explicit configuration required in the client.

During installation, listeners are created on nodes for the SCAN IP addresses. Oracle Net Services routes application requests to the least loaded instance providing the service. Because the SCAN addresses resolve to the cluster, rather than to a node address in the cluster, nodes can be added to or removed from the cluster without affecting the SCAN address configuration.

How GPnP Works: Cluster Node Startup

- IP addresses are negotiated for public interfaces using DHCP:
 - Node VIPs
 - SCAN VIPs
- A GPnP agent is started from the nodes Clusterware home.
- The GPnP agent gets its profile either locally or from one of the peer GPnP agents that responds.
- Shared storage is configured to match profile requirements.
- Service startup is specified in the profile, which includes:
 - Grid Naming Service for external names resolution
 - Single-client access name (SCAN) listener

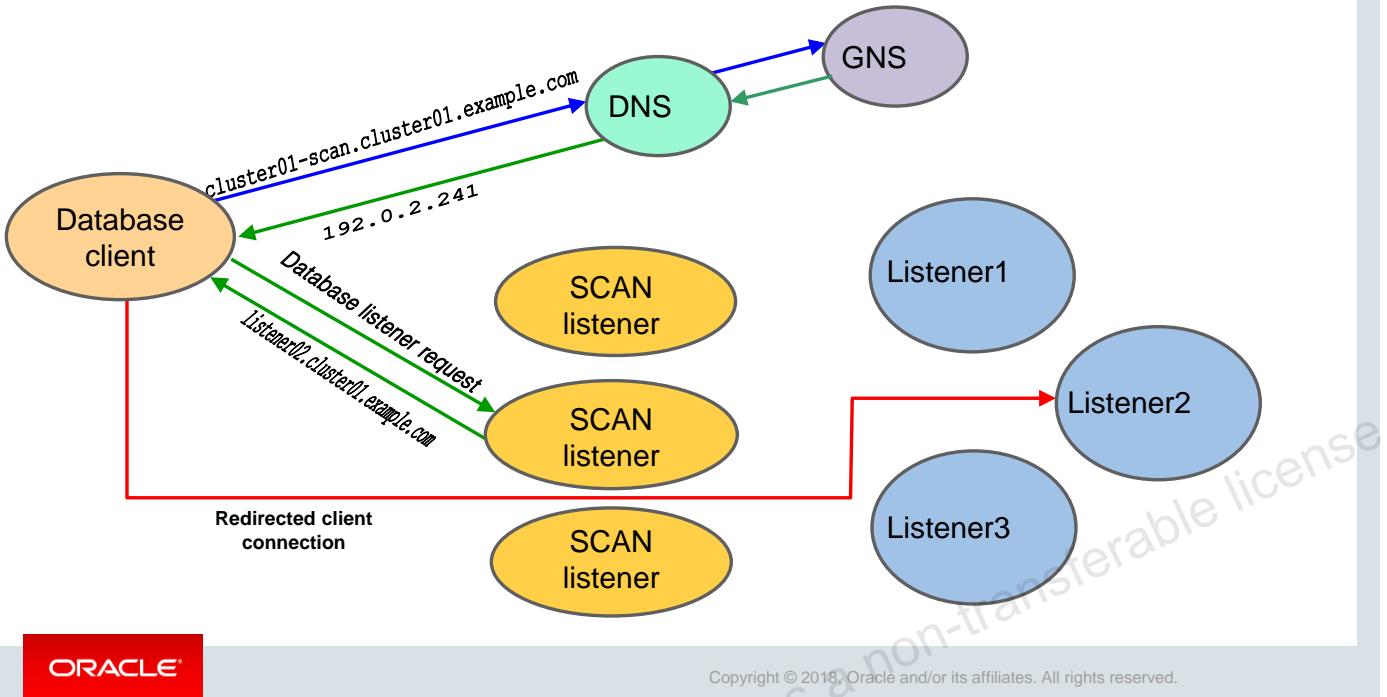


Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

When a node is started in a GPnP environment:

- Network addresses are negotiated for all interfaces using DHCP
- The Clusterware software on the starting node starts a GPnP agent
- The GPnP agent on the starting node gets its profile locally or uses resource discovery (RD) to discover the peer GPnP agents in the grid. If RD is used, it gets the profile from one of the GPnP peers that responds.
The GPnP agent acquires the desired network configuration from the profile. This includes creation of reasonable host names. If there are static configurations, they are used in preference to the dynamic mechanisms. Network interfaces may be reconfigured to match the profile requirements.
- Shared storage is configured to match the profile requirements
- System and service startup is done as configured in the image. In the case of RAC, the CSS and CRS systems will then be started, which will form the cluster and bring up appropriate database instances. The startup of services may run down their own placeholder values, or may dynamically negotiate values rather than rely on fixed-up configurations. One of the services likely to be started somewhere is the GNS system for external name resolution. Another of the services likely to be started is an Oracle SCAN listener.

Client Database Connections



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

In a GPNP environment, the database client no longer has to use the TNS address to contact the listener on a target node. Instead, it can use the EZConnect method to connect to the database. When resolving the address listed in the connect string, the DNS will forward the resolution request to the GNS with the SCAN VIP address for the chosen SCAN listener and the name of the database service that is desired. In EZConnect syntax, this would look like:

`scan-name.cluster-name.company.com/ServiceName`, where the service name might be the database name. The GNS will respond to the DNS server with the IP address matching the name given; this address is then used by the client to contact the SCAN listener. The SCAN listener uses its connection load-balancing system to pick an appropriate listener, whose name it returns to the client in an OracleNet Redirect message. The client reconnects to the selected listener, resolving the name through a call to the GNS.

The SCAN listeners must be known to all the database listener nodes and clients. The database instance nodes cross-register only with known SCAN listeners, also sending them per-service connection metrics. The SCAN known to the database servers may be profile data or stored in OCR.



Quiz

On OL 6 or RHEL 6, the `init.ohasd` entry in the `/etc/init/oracle-ohasd.conf` file is responsible for:

- a. Starting Oracle Clusterware when the node boots
- b. Mounting shared volumes as required by Oracle Clusterware
- c. Managing node evictions
- d. Restarting `ohasd` in the event of a crash



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to describe Oracle Clusterware:

- Architecture
- Startup details



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Practice 2: Overview

This practice covers the following topic:

- Introducing the laboratory environment for this course.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Unauthorized reproduction or distribution prohibited. Copyright© 2019, Oracle and/or its affiliates.

GANG LIU (gangl@baylorhealth.edu) has a non-transferable license
to use this Student Guide.

Cluster Configuration Options



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe available cluster types:
 - Standalone Cluster
 - Oracle Domain Services Cluster
 - Oracle Member Cluster for Oracle Databases
 - Oracle Member Cluster for Applications
 - Oracle Extended Clusters



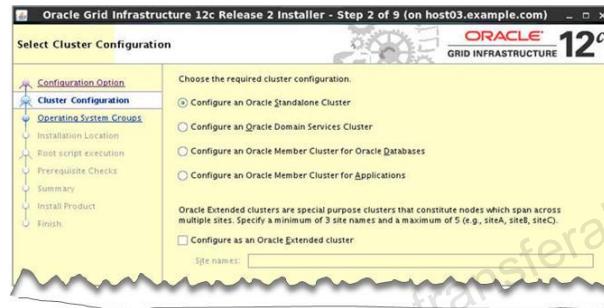
ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Cluster Configuration Options

The cluster configuration options that are available in Oracle Grid Infrastructure 12c, Release 2 include:

- Oracle Standalone Clusters
- Oracle Domain Services Cluster
- Oracle Member Cluster for Oracle Databases
- Oracle Member Cluster for Applications
- Oracle Extended Clusters



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Standalone Clusters

- Standalone Clusters contain two types of nodes: Hub Nodes and Leaf Nodes.
- A stand-alone cluster hosts all Grid Infrastructure services and ASM locally.
 - Hub nodes require direct access to shared storage.
- Standalone Clusters host the Grid Infrastructure Management Repository (GIMR) locally.
- When you deploy an Oracle Standalone Cluster, you can also choose to configure it as an Oracle Extended Cluster.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

In Oracle Clusterware 12c release 2 (12.2), all standalone clusters are configured as Oracle Flex Clusters, meaning that a cluster is configured with one or more Hub Nodes and can support a large number of Leaf Nodes. Clusters currently configured under older versions of Oracle Clusterware are converted in place as part of the upgrade process, including the activation of Oracle Flex ASM (which is a requirement for Oracle Flex Clusters).

An Oracle Standalone Cluster hosts all Oracle Grid Infrastructure services and Oracle ASM locally, and requires direct access to shared storage and contains two types of nodes arranged in a hub-and-spoke architecture: Hub Nodes and Leaf Nodes.

- The number of Hub Nodes in an Oracle Standalone Cluster can be as many as 64. The number of Leaf Nodes can be many more.
- The Hub Nodes and Leaf Nodes can host different types of applications.
- The Oracle Standalone Cluster Hub Nodes are tightly connected, and have direct access to shared storage. The Leaf Nodes do not require direct access to shared storage.
- The Hub Nodes can run in an Oracle Standalone Cluster configuration without having any Leaf Nodes as cluster member nodes, but Leaf Nodes must be members of a cluster with a pool of Hub Nodes.
- Shared storage is locally mounted on each of the Hub Nodes, with an Oracle ASM instance that is available to all Hub Nodes.

Oracle Standalone Clusters host Grid Infrastructure Management Repository (GIMR) locally. The GIMR is a multitenant database, which stores information about the cluster. This information includes the real-time performance data that the Cluster Health Monitor collects, and includes metadata required for Rapid Home Provisioning.

When you deploy an Oracle Standalone Cluster, you can also choose to configure it as an Oracle Extended Cluster. An Oracle Extended Cluster consists of nodes that are located in multiple locations or sites.

Oracle Cluster Domain

- An *Oracle Cluster Domain* is a new choice of deployment architecture for new clusters.
- Multiple cluster configurations are grouped under an Oracle Cluster Domain for:
 - Sharing the services that are available within the Oracle Cluster Domain
 - Ease of management
- The cluster configurations within the Oracle Cluster Domain include:
 - *Oracle Domain Services Cluster*
 - *Oracle Member Clusters*
- Cluster Domains enable you to standardize, centralize, and optimize your RAC deployment for the private database cloud.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

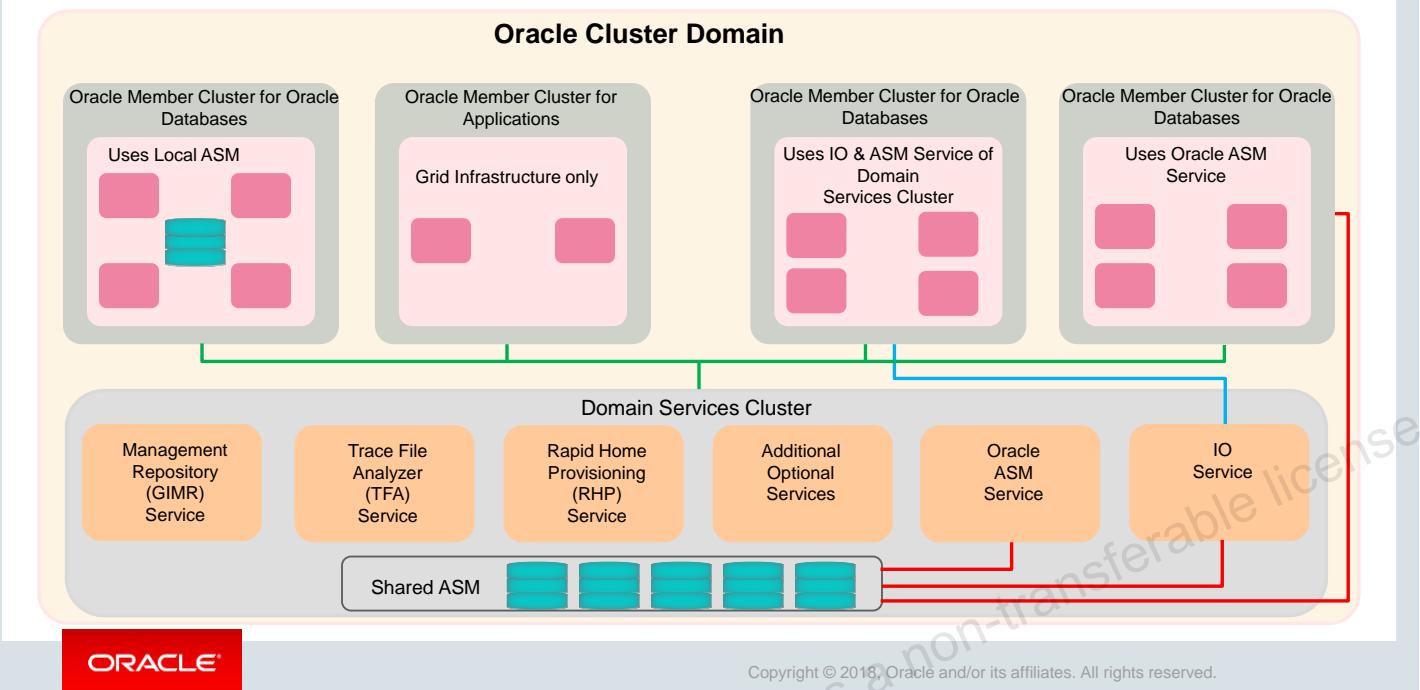
An Oracle Cluster Domain is a choice of deployment architecture for new clusters, introduced in Oracle Clusterware 12c, Release 2.

Oracle Cluster Domain enables you to standardize, centralize, and optimize your Oracle Real Application Clusters (Oracle RAC) deployment for the private database cloud. Multiple cluster configurations are grouped under an Oracle Cluster Domain for management purposes, and use the shared services that are available within that Oracle Cluster Domain. The cluster configurations within that Oracle Cluster Domain include Oracle Domain Services Cluster and Oracle Member Clusters.

Oracle Domain Services Cluster

Architecture

Private Network
SAN Storage
ASM Storage Network



The Oracle Domain Services Cluster provides centralized services to other clusters within the Oracle Cluster Domain. These services include:

- A centralized Grid Infrastructure Management Repository (housing the MGMTDB for each of the clusters within the Oracle Cluster Domain)
- Trace File Analyzer (TFA) services, for targeted diagnostic data collection for Oracle Clusterware and Oracle Database
- Consolidated Oracle ASM storage management service
- An optional Rapid Home Provisioning (RHP) service to install clusters, and provision, patch, and upgrade Oracle Grid Infrastructure and Oracle Database homes. When you configure the Oracle Domain Services Cluster, you can also choose to configure the Rapid Home Provisioning Server.

An Oracle Domain Services Cluster provides these centralized services to Oracle Member Clusters. Oracle Member Clusters use these services:

- For centralized and simplified management
- To reduce their local resource usage and dependence

Oracle Member Clusters

- Member Clusters use centralized services from the Domain Services Cluster and can host databases or applications.
- Oracle Member Clusters do not need direct connectivity to shared disks.
- Using the shared ASM service, they can connect to the IO Service to access a centrally managed pool of storage.
- To use shared ASM services from the Domain Services Cluster, the member cluster needs connectivity to the ASM networks of the Oracle Domain Services Cluster.
- Member Clusters can be either of the following:
 - *Oracle Member Clusters for Oracle Databases*
 - *Oracle Member Clusters for Applications*



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Member Clusters use centralized services from the Oracle Domain Services Cluster, and can host databases or applications. Oracle Member Clusters can be of two types: Oracle Member Clusters for Oracle databases or Oracle Member Clusters for applications.

Oracle Member Clusters do not need direct connectivity to shared disks in the Oracle Domain Services Cluster. Using the shared Oracle ASM service, they can use network connectivity to the IO Service to access a centrally managed pool of storage. To use shared Oracle ASM services from the Oracle Domain Services Cluster, the member cluster needs connectivity to the Oracle ASM networks of the Oracle Domain Services Cluster.

Oracle Member Clusters cannot provide services to other clusters. For example, you cannot configure and use a member cluster as a GNS server or Rapid Home Provisioning Server.

Oracle Member Cluster for Oracle Databases

- An Oracle Member Cluster for Oracle Databases supports:
 - Oracle Real Application Clusters
 - Oracle RAC One Node database instances
- This type of cluster:
 - Registers with the management repository service
 - Uses the centralized TFA service
- An Oracle Member Cluster for Oracle Databases can:
 - Be configured with local Oracle ASM storage management
 - Make use of the consolidated Oracle ASM storage management service offered by the Oracle Domain Services Cluster



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

An Oracle Member Cluster for Oracle Databases supports Oracle Real Application Clusters (Oracle RAC) or Oracle RAC One Node database instances. This cluster registers with the management repository service and uses the centralized TFA service. It can use additional services as needed. An Oracle Member Cluster for Oracle Databases can be configured with local Oracle ASM storage management or use the consolidated Oracle ASM storage management service offered by the Oracle Domain Services Cluster.

An Oracle Member Cluster for Oracle Databases always uses the remote Grid Infrastructure Management Repository (GIMR) from its Oracle Domain Services Cluster. For two-node or four-node clusters, hosting the GIMR on a remote cluster reduces the overhead of running an extra infrastructure repository on a cluster.

Oracle Member Cluster for Applications

- Oracle Member Cluster for Applications hosts applications other than Oracle Databases, as part of an Oracle Cluster Domain.
- An Oracle Member Cluster requires connectivity to Oracle Cluster Domain Services for centralized management.
- Oracle Member Clusters use remote Oracle ASM storage.
- This type of cluster configuration enables high availability of any software application.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Member Cluster for Applications hosts applications other than Oracle Database, as part of an Oracle Cluster Domain. An Oracle Member Cluster requires connectivity to Oracle Cluster Domain Services for centralized management and resource efficiency. The Oracle Member Cluster uses remote Oracle ASM storage and does not require direct shared storage access. This cluster configuration enables high availability of any software application.

Member Cluster Manifest File for Member Clusters

- Create a Member Cluster Manifest file to specify the Oracle Member Cluster configuration for:
 - Grid Infrastructure Management Repository
 - Grid Naming Service
 - Oracle ASM storage server
 - Rapid Home Provisioning
1. If the Member Cluster accesses direct or indirect ASM storage, then enable access to the disk group.

```
SQL> ALTER DISKGROUP data SET ATTRIBUTE 'access_control.enabled' = 'true';
```

```
[grid@host01 bin] $ cd /u01/app/12.2.0/grid/bin
[grid@host01 bin] $ ./crsctl create member_cluster_configuration
$HOME/cluster02 -file cluster01_manifest.xml -member_type database
-domain_services asm_storage indirect rhp
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

If you are creating a Member Cluster, you will need to generate a Member Cluster Manifest file. Create a Member Cluster Manifest file to specify the Oracle Member Cluster configuration for the Grid Infrastructure Management Repository (GIMR), Grid Naming Service, Oracle ASM storage server, and Rapid Home Provisioning.

Oracle Member Clusters use Oracle ASM storage from the Oracle Domain Services Cluster. A Grid Naming Service (GNS) without zone delegation must be configured so that the GNS virtual IP address (VIP) is available for connection.

1. If the Member Cluster accesses direct or indirect ASM storage, enable access to the disk group. Connect to any Oracle ASM instance as the SYSASM user and run the following command:


```
ALTER DISKGROUP dg_name SET ATTRIBUTE 'access_control.enabled' = 'true';
```
2. From the Grid home on the Domain Services Cluster, create the Member Cluster Manifest file:


```
$ cd Grid_home/bin
$ ./crsctl create member_cluster_configuration member_cluster_name
-file file_name -member_type database|application [-version
member_cluster_ver [-domain_services [asm_storage
local|direct|indirect] [rhp]]]
```

`-file` specifies the full path of the XML file to export the credentials and `-version` is the five-digit Client Cluster version, for example, 12.2.0.1.0, if it is different from the Storage Server version. The Storage Server version is used if `-version` is not specified.

In the options for `-domain_services`, specifying `rhp` generates credentials and configuration for an RHP Client Cluster, and specifying `asm_storage` generates credentials and configuration for an Oracle ASM Client Cluster. If specified, `direct` signifies direct storage access; otherwise indirect access is used.

This command creates a member cluster manifest file that contains configuration details about the Grid Infrastructure Management Repository (GIMR), storage services, and Rapid Home Provisioning for the Oracle Member Cluster.

Member Cluster Manifest File for Member Clusters

3. Export GNS client data to the manifest file created earlier.

```
[grid@host01 bin]$ srvctl export gns -clientdata  
$HOME/cluster01_manifest.xml -role CLIENT
```

4. Copy the file to a location on the Member Cluster and select the file during the installation of the Member Cluster.

```
[grid@host01 bin] $ scp cluster01_manifest.xml host03:/home/grid
```



3. GNS client data is required if the Oracle Member Cluster uses dynamic networks and the server cluster has GNS with zone delegation. Provide the GNS client data as follows:

As the `root` or `grid` user, export the Grid Naming Service (GNS) client data to the member cluster manifest file created earlier:

```
$ srvctl export gns -clientdata manifest_file_name -role CLIENT
```

The GNS configuration is appended to the member cluster manifest file.

4. Copy the manifest file to a location on the Oracle Member Cluster, and select the file during the installation and configuration of the Oracle Member Cluster.

Oracle Extended Clusters

- An Oracle Extended Cluster consists of nodes that are located in multiple locations or sites.
- When you deploy an Oracle Standalone Cluster, you can also choose to configure the cluster as an Extended Cluster.
- You can extend a cluster across two, or more, geographically separate sites, each equipped with its own storage.
 - If one of the sites fails, the other site acts as an active standby.
- You can configure an Oracle Extended Cluster:
 - **Option 1:** When you install Oracle Grid Infrastructure
 - **Option 2:** Post-installation by using the `ConvertToExtended` script
- Extended Clusters are managed by using `crsctl`.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Starting with Oracle Grid Infrastructure 12c Release 2 (12.2), the *Extended Distance Cluster* has advanced to a first class feature, which is *Oracle Extended Cluster*. It uses Oracle ASM for data management.

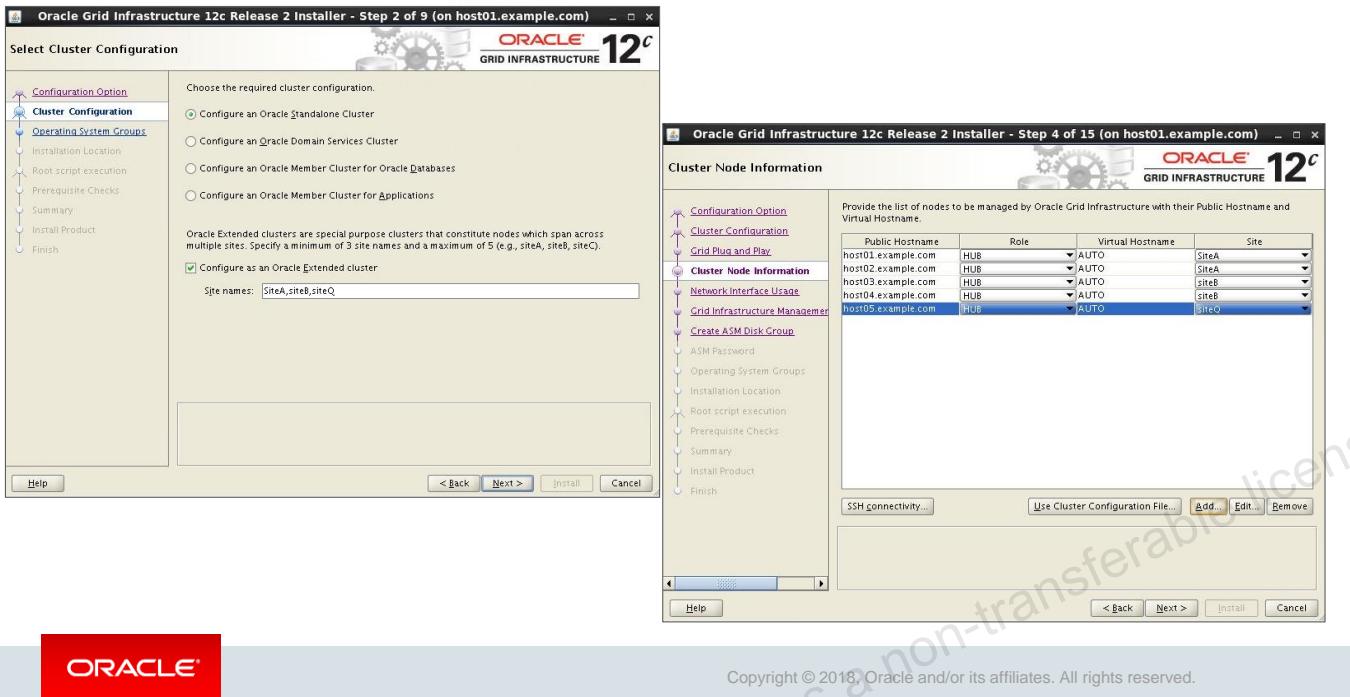
An Oracle Extended Cluster consists of nodes that are located in multiple locations called sites. When you deploy an Oracle Standalone Cluster, you can also choose to configure the cluster as an Oracle Extended Cluster. You can extend an Oracle RAC cluster across two, or more, geographically separate sites, each equipped with its own storage. In the event that one of the sites fails, the other site acts as an active standby.

Both Oracle ASM and the Oracle Database stack, in general, are designed to use enterprise-class shared storage in a data center. Fibre Channel technology, however, enables you to distribute compute and storage resources across two or more data centers, and connect them through Ethernet cables and Fibre Channel, for compute and storage needs, respectively.

You can configure an Oracle Extended Cluster when you install Oracle Grid Infrastructure. You can also do so post-installation by using the `ConvertToExtended` script. You manage your Oracle Extended Cluster by using `crsctl`.

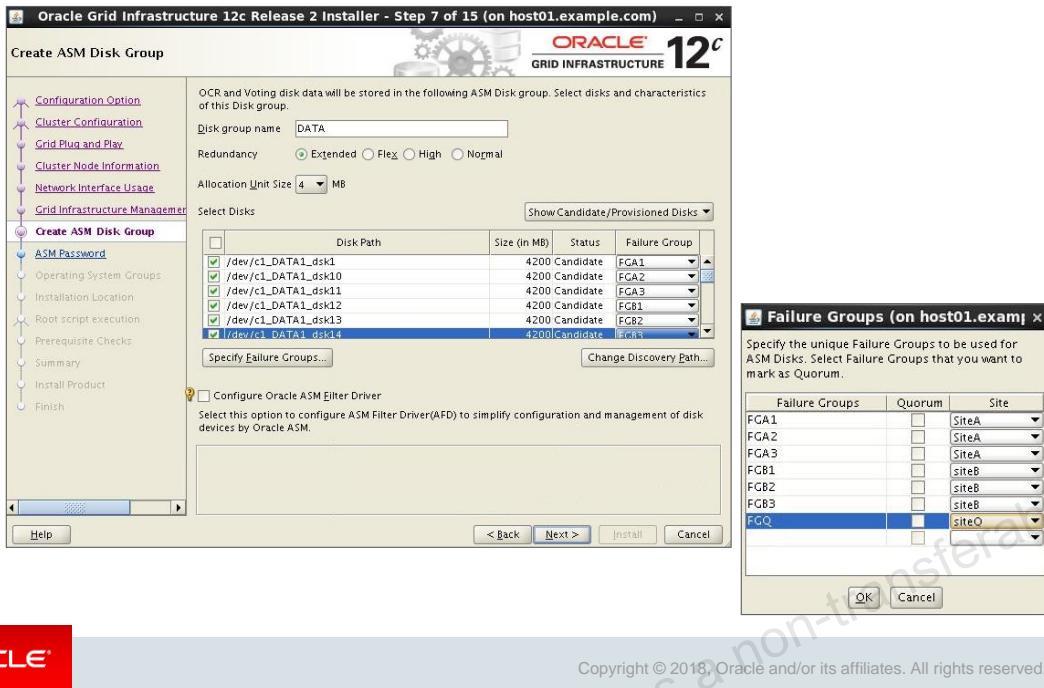
Oracle recommends that you deploy Oracle Extended Clusters with normal redundancy disk groups. You can assign nodes and failure groups to sites. Sites contain failure groups, and failure groups contain disks. For normal redundancy disk groups, a disk group provides one level of failure protection, and can tolerate the failure of either a site or a failure group.

Option 1: Configure an Oracle Extended Cluster



When you deploy an Oracle Standalone Cluster, you can also choose to configure the cluster as an Extended Cluster. The example in the slide shows how to configure an Oracle Extended cluster that is composed of three sites (Site A, Site B, and Site Q) when you install Oracle Grid Infrastructure.

Assign Failure Groups to Sites



The example in the slide shows how to define the failure group names and assign to sites.

- Select **Extended** redundancy
- Click on the **Specify Failure Groups** button to enter failure group and site information for all the failure groups.
- Click on the **Change Discovery Path** button to display the candidate ASM disks.
- Configure the candidate ASM disks to assign to the proper failure groups.

Option 2: Configure Oracle Extended Clusters

1. Use `crsctl` to determine a cluster's extended status.

```
$ crsctl get cluster extended
CRS-6579: The cluster is 'NOT EXTENDED'

$ crsctl query cluster site -all
Site 'cluster01' identified by 7b7b3bef4c1f5ff9ff8765bceb45433a'
in state 'ENABLED', band contains nodes 'host01,host02,host03,host04', and disks.
```

2. As root, log in to the first node, and run the following command:

```
# rootcrs.pl -converttoextended -first -sites list_of_sites -site
node_site
```

3. As root, on all other nodes, run the following command:

```
# rootcrs.pl -converttoextended -site node_site
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Extended Cluster enables you to deploy Oracle RAC databases on a cluster, in which some of the nodes are located in different sites. However, before starting, ensure that you have upgraded to Oracle Grid Infrastructure 12c Release 2.

You can also configure an Oracle Extended Cluster using the `ConvertToExtended` script. Using the script, you can create multiple data sites and associate a node with each data site. All Flex ASM storage remains associated with the default cluster site because there is no mechanism to convert an existing disk group to an extended disk group.

After you convert your cluster to an Extended Cluster, the voting file membership remains flat, not hierarchical. You must also add an extended disk group and migrate the voting files to the extended disk group to take advantage of the site-specific hierarchical voting file algorithm.

1. Use `crsctl` to query the cluster to determine its extended status:

- \$ crsctl get cluster extended
- \$ crsctl query cluster site -all

2. As the root user, log in to the first node, and run the following command:

```
rootcrs.pl -converttoextended -first -sites list_of_sites -site
node_site
```

Where `list_of_sites` is the comma-separated list of sites in the extended cluster, and `node_site` is the node containing the site.

```
[root@host01 ~]# cd /u01/app/12.2.0/grid/crs/install
[root@host01 install]# ./rootcrs.pl -converttoextended -first -sites
newyork,newjersey -site newjersey
```

```
Using configuration parameter file: ./crsconfig_params
The log of current session can be found at:
  /u01/app/grid/crsdata/host01/crsconfig/rootcrs_host01_2017-12-
  19_12-27-19AM.log
CRS-4123: Oracle High Availability Services has been started.
```

3. As the root user, on all other nodes, run the following command:

```
# rootcrs.pl -converttoextended -site node_site
[root@host01 install]# ssh host02
/u01/app/12.2.0/grid/crs/install/rootcrs.pl -converttoextended -
site newjersey
```

Using configuration parameter file:
/u01/app/12.2.0/grid/crs/install/crsconfig_params

The log of current session can be found at:

```
  /u01/app/grid/crsdata/host02/crsconfig/rootcrs_host02_2017-12-
  19_01-07-56PM.log
```

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

```
[root@host01 install]# ssh host03
/u01/app/12.2.0/grid/crs/install/rootcrs.pl -converttoextended -
site newyork
```

Using configuration parameter file:
/u01/app/12.2.0/grid/crs/install/crsconfig_params

The log of current session can be found at:

```
  /u01/app/grid/crsdata/host03/crsconfig/rootcrs_host03_2017-12-
  19_02-05-05PM.log
```

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

```
[root@host01 install]# ssh host04
/u01/app/12.2.0/grid/crs/install/rootcrs.pl -converttoextended -
site newyork
```

Using configuration parameter file:
/u01/app/12.2.0/grid/crs/install/crsconfig_params

The log of current session can be found at:

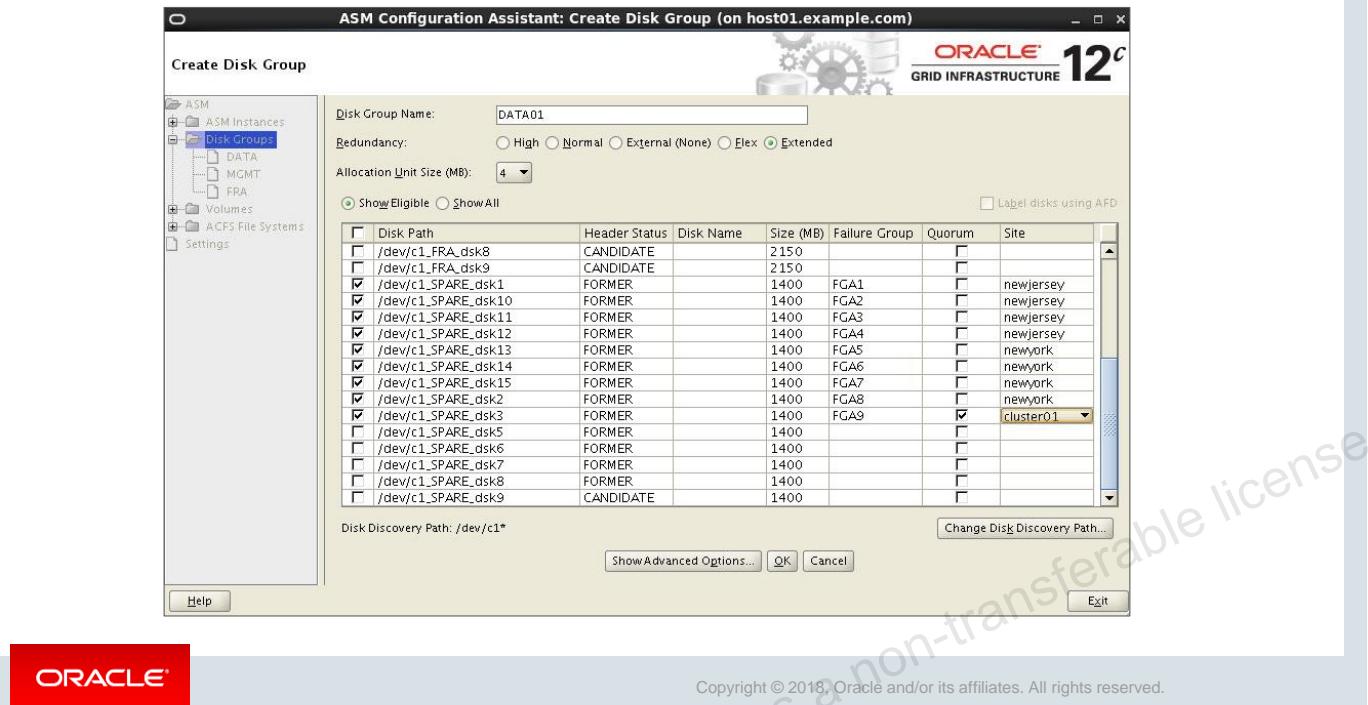
```
  /u01/app/grid/crsdata/host04/crsconfig/rootcrs_host04_2017-12-
  19_02-07-34PM.log
```

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

4. Determine the cluster's extended status.

```
[root@host01 install]# crsctl get cluster extended
CRS-6577: The cluster is extended.
```

Assign Failure Groups to Sites Using ASMCA



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

(Continued from the previous slide)

5. Assign ASM storage (Failure Groups) to sites. The example in the slide shows how to define the failure group names and assign to sites using ASMCA.
6. Optional: Delete the default site after the associated nodes and storage have been migrated.

```
[root@host01 install]# crsctl query cluster site -all
Site 'cluster01' identified by GUID 'a99aee2d50b17f20ffaee1f8ba4df287' in
state 'ENABLED' contains disks 'DATA01_0008' and no nodes .

Site 'newjersey' identified by GUID 'a053f55e812d7f5dff1d5dedd5f6d5b0' in
state 'ENABLED' contains nodes 'host01,host02' and disks
'DATA01_0000,DATA01_0001,DATA01_0002,DATA01_0003'.

Site 'newyork' identified by GUID 'ba9174b9d2a2cf5aff6d6030b23070af' in
state 'ENABLED' contains nodes 'host03,host04' and disks
'DATA01_0004,DATA01_0005,DATA01_0006,DATA01_0007'.

[root@host01 install]# crsctl delete cluster <site_name>
```



Quiz

Which of the following cluster types are available in Oracle Clusterware 12.2?

- a. Standalone Cluster
- b. Oracle Domain Services Cluster
- c. Oracle Leaf Cluster
- d. Oracle Member Cluster for Oracle Databases
- e. Oracle Member Cluster for Applications
- f. Oracle Extended Clusters



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Describe Available Cluster Types
 - Standalone Cluster
 - Oracle Domain Services Cluster
 - Oracle Member Cluster for Oracle Databases
 - Oracle Member Cluster for Applications
 - Oracle Extended Clusters



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Unauthorized reproduction or distribution prohibited. Copyright© 2019, Oracle and/or its affiliates.

GANG LIU (gangl@baylorhealth.edu) has a non-transferable license
to use this Student Guide.

4

Grid Infrastructure Preinstallation Tasks



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to perform preinstallation tasks for Grid Infrastructure.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Preinstallation Planning



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Shared Storage Planning for Grid Infrastructure and RAC

| Storage Option | OCR and Voting Disks | Grid Home | RAC Home | RAC DB files | Oracle Recovery Files |
|-------------------|----------------------|-----------|-------------------------------------|------------------|-----------------------|
| ASM | Yes | No | No | Yes | Yes |
| ACFS | No | No | Yes ² No ³ | Yes ⁴ | Yes ⁴ |
| NFS | No ¹ | Yes | Yes | Yes | Yes |
| Raw Devices | No | No | No | No | No |
| Local File System | No | Yes | Yes | No | No |

1. Direct Shared File System Placement for Oracle Cluster Registry (OCR) and Voting Files is not supported in 12c Release 2.
2. Yes for Oracle Database 11g Release 2, or more recent releases
3. No for running Oracle Database on Leaf Nodes because ACFS cannot run on Leaf Nodes
4. Yes for Oracle Database 12c Release 1 and later



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Using a Shared File System with Grid Infrastructure

- To use an NFS file system, it must be on a supported NAS device.
 - Check My Oracle Support: <https://support.oracle.com>
- If you need to place the OCR and Voting Files on a shared file system:
 - You must create Oracle ASM disks on supported network file systems that you plan to use for hosting Oracle Clusterware files before installing Oracle Grid Infrastructure.
 - You can then use the Oracle ASM disks in an Oracle ASM disk group to manage Oracle Clusterware files.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

To use a shared file system for Oracle Clusterware, Oracle ASM, and Oracle RAC, the file system must comply with the following requirements:

- To use an NFS file system, it must be on a supported NAS device. Log in to My Oracle Support and click Certifications to find the most current information about supported NAS device.

If you need to place your Oracle Cluster Registry (OCR) and Voting Files on a shared file system, note that:

- Starting with Oracle Grid Infrastructure 12c Release 2 (12.2), the placement of Oracle Clusterware files: the Oracle Cluster Registry (OCR), and the Voting Files, directly on a shared file system is unsupported in favor of having Oracle Clusterware files managed by Oracle Automatic Storage Management (Oracle ASM). You cannot place Oracle Clusterware files directly on a shared file system.
- If you need to use a supported shared file system, either a Network File System, or a shared cluster file system instead of native disk devices, then you must create Oracle ASM disks on supported network file systems that you plan to use for hosting Oracle Clusterware files before installing Oracle Grid Infrastructure. You can then use the Oracle ASM disks in an Oracle ASM disk group to manage Oracle Clusterware files.

If your Oracle Database files are stored on a shared file system, then you can continue to use shared file system storage for database files, instead of moving them to Oracle ASM storage.

Logical Volume Managers and Grid Infrastructure

- Grid Infrastructure and Oracle RAC support only cluster-aware volume managers.
- Some third-party volume managers are not cluster-aware and are therefore not supported.
- To confirm that a volume manager is supported, click Certifications on My Oracle Support:
<https://support.oracle.com>



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Managing Voting Disks in ASM

- Each node must be able to access a majority of voting disks; otherwise, it will be evicted from the cluster.
- Voting disks must be stored on an ASM disk group.
 - They are not regular ASM files.
 - Clusterware knows the location in case ASM is unavailable.
- The number of voting disks is determined by the ASM disk group redundancy setting.
 - 1 voting disk for external redundancy disk group
 - 3 voting disks for normal redundancy disk group
 - 5 voting disks for high redundancy disk group
- A separate failure group is required for each voting disk.
- Voting disks are managed using the `crsctl` utility.

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware uses voting disk files, also called voting disks, to determine which nodes are members of a cluster and to maintain the integrity of the cluster. When you configure voting disks on ASM, you do not need to manually configure the voting disks. Depending on the redundancy of the specified disk group, a predefined number of voting disks are created.

ASM manages voting disks differently from other files that it stores. When you initially configure Oracle Clusterware, you specify the disk group to contain the voting disks. Each voting disk is housed in a separate ASM failure group. You must specify enough failure groups to support the number of voting disks associated with each disk group redundancy setting. For example, you must have at least three failure groups to store your voting disks in a normal redundancy disk group. Voting disks do not appear as regular files within ASM, rather Clusterware records exactly where the voting disk information is located. This arrangement exists so that if ASM is unavailable for any reason, Cluster Synchronization Services can still access the voting disks and maintain the cluster.

One of the benefits of using an ASM disk group, with either normal or high redundancy, is that if a disk containing a voting disk fails, as long as there is another disk available in the disk group, ASM will automatically recover the voting disk. Voting disks are managed by using the `crsctl` utility. For example, the following command migrates voting disks from their current location to an ASM disk group named `VOTE`:

```
# crsctl replace votedisk +VOTE
```

Sizing Storage for Oracle Standalone Cluster

| Cluster Configuration | Redundancy Level | Space Required for DATA Disk Group containing Oracle Clusterware Files (OCR and Voting Files) | Space Required for MGMT Disk Group Containing the GIMR and Oracle Clusterware Backup Files | Total Storage |
|---|------------------|---|--|---------------|
| Two nodes, 4MB Allocation Unit (AU), one Oracle ASM disks | External | 1.4 GB | At least 37.6 GB for a cluster with 4 nodes or less. Additional 4.7 GB space required for clusters with 5 or more nodes | 39 GB |
| Two nodes, 4MB Allocation Unit (AU), three Oracle ASM disks | Normal | 2.5 GB | 75.5 GB | 78 GB |
| Two nodes, 4MB Allocation Unit (AU), five Oracle ASM disks | High | 3.6 GB | 113.4 GB | 117 GB |
| Two nodes, 4MB Allocation Unit (AU), three Oracle ASM disks | Flex | 2.5 GB | 75.5 GB | 78 GB |



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The slide shows the information on the minimum number of disks and the minimum disk space requirements based on the redundancy type, for installing Oracle Clusterware files, and installing the starter database, for various Oracle Cluster deployments.

During installation of an Oracle Standalone Cluster, if you create the MGMT disk group for Grid Infrastructure Management Repository (GIMR), then the installer requires that you use a disk group with at least 35 GB of available space.

The Grid Infrastructure Management Repository (GIMR) is a multitenant database with a pluggable database (PDB) for the GIMR of each cluster. The GIMR stores the following information about the cluster:

- Real time performance data the Cluster Health Monitor collects
- Fault, diagnosis, and metric data the Cluster Health Advisor collects
- Cluster-wide events about all resources that Oracle Clusterware collects
- CPU architecture data for Quality of Service Management (QoS)
- Metadata required for Rapid Home Provisioning

Based on the cluster configuration you want to install, the Oracle Clusterware space requirements vary for different redundancy levels. The following tables list the space requirements for each cluster configuration.

For more information about storage space requirements for Oracle Domain Service Clusters and member clusters, refer to *Oracle Grid Infrastructure Installation and Upgrade Guide 12c Release 2 (12.2)*.

GIMR Configuration Details

| PARAMETER | 12.1.0.2 | 12.2.0.1 Standalone | 12.2.0.1 DCS |
|----------------------|----------|---------------------|--------------|
| INSTANCE_NAME | -MGMTDB | -MGMTDB | -MGMTDB |
| DB_NAME | _MGMTDB | _MGMTDB | _MGMTDB |
| SGA_MAX_SIZE | 752MB | 1GB | 4GB |
| PGA_AGGREGATE_TARGET | 352MB | 500MB | 2GB |
| PGA_AGGREGATE_LIMIT | 2GB | 2GB | 6GB |
| CPU_COUNT | 2 | 2 | 8 |
| PROCESSES | 300 | 500 | 2000 |
| SESSIONS | 472 | 772 | 3024 |
| USE_LARGE_PAGES | TRUE | TRUE | TRUE |



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The slide shows the Grid Infrastructure Management Repository (GIMR) configuration details.



Quiz

The Oracle Cluster Registry (OCR) must be stored in ASM.

- a. True
- b. False



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

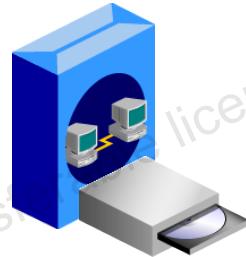
Grid Infrastructure Preinstallation Tasks



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Grid Infrastructure 12c Installation

1. Check system requirements.
2. Check network requirements.
3. Install required operating system packages.
4. Set kernel parameters.
5. Create groups and users.
6. Create required directories.
7. Configure installation owner shell limits.
8. Install Grid Infrastructure.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

To successfully install Oracle Grid Infrastructure, it is important that you have an understanding of the tasks that must be completed and the order in which they must occur. Before the installation can begin in earnest, each node that is going to be part of your cluster installation must meet the hardware and software requirements that are covered in this lesson. You must perform step-by-step tasks for hardware and software verification, as well as for platform-specific preinstallation procedures. You must install the operating system patches required by the cluster software and verify that the kernel parameters are correct for your needs.

Oracle Grid Infrastructure must be installed by using the graphical OUI. Character-based tool installations are not possible; however, OUI can be run in silent mode using response files to supply the values that the installation would need. Ensure that your cluster hardware is functioning normally before you begin this step. Failure to do so results in an aborted or nonoperative installation.

If you intend to use Enterprise Manager Cloud Control to manage your cluster deployments, you must next install the Enterprise Manager (EM) agent on each cluster node.

Note: This lesson provides details about performing an installation, but it should not be used as a substitute for the Installation manual for your platform.

General Server Minimum Requirements

- Select servers with the same instruction set architecture.
- Running 32-bit and 64-bit Oracle software versions in the same cluster stack is not supported.
- Make sure the servers are started with run level 3 or 5.
- Ensure that display cards provide at least 1024 x 768 display resolution.
- Ensure that servers run the same operating system binary.
- Cluster nodes can have CPUs of different speeds or sizes, but they should have the same hardware configuration.
 - If you configure clusters using different configuration, you should categorize cluster nodes into homogenous pools.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

When selecting and configuring servers for your cluster, it is important to remember the following general guidelines and minimum requirements:

- Select servers with the same instruction set architecture; running 32-bit and 64-bit Oracle software versions in the same cluster stack is not supported.
- Ensure that the server is started with run level 3 or 5 (Linux).
- Ensure that display cards provide at least 1024 x 768 display resolution, so that OUI displays correctly while performing a system console-based installation.
- Ensure that servers run the same operating system binary. Oracle Grid Infrastructure installations and Oracle Real Application Clusters (Oracle RAC) support servers with different hardware in the same cluster.
- Your cluster can have nodes with CPUs of different speeds or sizes, but Oracle recommends that you use nodes with the same hardware configuration. Oracle recommends that if you configure clusters by using different configuration, then you categorize cluster nodes into homogenous pools as part of your server categorization management policy.

Checking System Requirements

- At least 8 GB of physical memory is needed.
- Set swap space equal to RAM for systems with 4 to 16 GB of RAM; set swap space to 16 GB for larger systems.

```
[root@host01 ~]# free -m
total    used     free   shared   buffers   cached
Mem:      4458  3323    1135       0      73    741
 -/+ buffers/cache:  2507    951
Swap:     8636  214    8422
```

- The local /tmp directory should have at least 1 GB free.
- At least 12 GB of local storage is needed for the software.

```
[root@host01 ~]# df -h /tmp /u01
Filesystem           Size  Used Avail Use% Mounted on
/dev/xvda2            12G  2.9G  7.8G  27% /
/dev/xvdb1            30G  7.5G  21G  27% /u01
```

- Shared memory must be greater than the sum of the SGA and the PGA of the databases on the server.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The system must meet the following minimum hardware requirements:

- The minimum required RAM is 8 GB for Grid Infrastructure. To determine the amount of physical memory, enter the following command, use the `free -m` command.
- The minimum required swap space should be equal to RAM for systems with 4 to 16 GB of RAM, and should be 16 GB for systems with more than 16 GB of RAM. To determine the size of the configured swap space, use the `free -m` command.
- At least 1 GB of disk space must be available in the /tmp directory (`TMP` and `TMPDIR` variables can force another location to be used)..
- At least 12 GB is required on each node for the Grid Infrastructure software home.
- At least 9GB for Oracle Database Enterprise Edition

If you intend to install Oracle Databases or Oracle RAC databases on the cluster, be aware that the size of the shared memory mount area (`/dev/shm`) on each server must be greater than the system global area (SGA) and the program global area (PGA) of the databases on the servers. Review expected SGA and PGA sizes with database administrators to ensure that you do not have to increase `/dev/shm` after databases are installed on the cluster.

Note: With the introduction of Oracle Database 12c, 32-bit systems are no longer supported.

Enabling the Name Service Cache Daemon (nscd)

- To allow Clusterware to better tolerate network failures when using NAS or NFS storage, enable nscd:

```
# /sbin/service nscd start
```

- To check to see if nscd is set to load on system startup, use the `chkconfig` command:

```
# chkconfig --list nscd
nscd 0:off 1:off 2:off 3:on 4:off 5:off 6:off
```

- It should be “on” for run levels 3 and 5.
- To alter the preceding configuration to ensure nsqd is on for both run levels, execute the following command:

```
# chkconfig --level 35 nscd on
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

To allow Oracle Clusterware to better tolerate network failures with NAS devices or NFS mounts, enable the Name Service Cache Daemon (nscd). The nsqd provides a caching mechanism for the most common name service requests.

To check to see if nsqd is set to load when the system is restarted, enter the command `chkconfig --list nscd`. For example:

```
# chkconfig --list nscd
```

In the example in the slide, nsqd is enabled for run level 3, and disabled for run level 5. The nsqd should be enabled for both run level 3 and run level 5.

To change the configuration to ensure that nsqd is enabled for both run level 3 and run level 5, enter one of the following command as root:

```
# chkconfig --level 35 nscd on
```

For Oracle Linux 7, verify that nsqd is enabled as follows:

```
systemctl --all |grep nscd
```

```
nscd.service loaded active running Name Service Cache Daemon
```

Setting the Disk I/O Scheduler on Linux

- Disk I/O schedulers reorder, delay, or merge requests for disk I/O to achieve better throughput and lower latency.
- Linux has multiple disk I/O schedulers available:
 - Deadline
 - Noop
 - Anticipatory
 - Completely Fair Queuing (CFQ)
- For best performance for ASM, Oracle recommends that you use the Deadline I/O scheduler.
- On each cluster node, enter the following command to ensure that the Deadline disk I/O scheduler is configured:

```
# echo deadline > /sys/block/{ASM_DISK}/queue/scheduler
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Disk I/O schedulers reorder, delay, or merge requests for disk I/O to achieve better throughput and lower latency. Linux has multiple disk I/O schedulers available, including Deadline, Noop, Anticipatory, and Completely Fair Queuing (CFQ). For best performance for Oracle ASM, Oracle recommends that you use the Deadline I/O scheduler.

On each cluster node, enter the following command to ensure that the Deadline disk I/O scheduler is configured for use:

```
# echo deadline > /sys/block/${ASM_DISK}/queue/scheduler
```

To make the I/O Scheduler selection persistent across system reboots, append an `elevator` statement similar to the following to the kernel line in the `/boot/grub/grub.conf` file.

```
kernel /vmlinuz-2.6.39-400.17.2.el6uek.x86_64 ro root=LABEL=root ... selinux=0  
elevator=deadline
```

Cluster Name and SCAN Requirements

- Cluster Name must meet the following requirements:
 - The cluster name is case-insensitive, must be unique across your enterprise, must be at least one character long and no more than 15 characters in length, must be alphanumeric, cannot begin with a numeral, and may contain hyphens (-). Underscore characters (_) are not allowed.
- SCAN must meet the following requirements:
 - The SCAN and cluster name are entered in separate fields during installation, so cluster name requirements do not apply to the name used for the SCAN, and the SCAN can be longer than 15 characters.
 - If you enter a domain with the SCAN name, and you want to use GNS with zone delegation, then the domain must be the GNS domain.

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Cluster Name and SCAN must meet the following requirements:

- The cluster name is case-insensitive, must be unique across your enterprise, must be at least one character long and no more than 15 characters in length, must be alphanumeric, cannot begin with a numeral, and may contain hyphens (-). Underscore characters (_) are not allowed.
- The SCAN and cluster name are entered in separate fields during installation, so cluster name requirements do not apply to the name used for the SCAN, and the SCAN can be longer than 15 characters. If you enter a domain with the SCAN name, and you want to use GNS with zone delegation, then the domain must be the GNS domain.

Note: Select your cluster name carefully. After installation, you can only change the cluster name by reinstalling Oracle Grid Infrastructure.

Checking Network Requirements

- Each node must have at least two NICs.
- Interface names must be the same on all nodes.
- Public NIC supports TCP/IP; private NIC supports UDP.
- IP addresses are configured using one of the following options:
 - Oracle Grid Naming Service (GNS) using one static address defined during installation
 - Static addresses that network administrators assign on a network domain name server (DNS) for each node
- Public IP must be registered in the domain name server (DNS) or the /etc/hosts file.

```
# cat /etc/hosts
##### Public Interfaces - net0 #####
xxx.xxx.100.11  host01.example.com  host01
xxx.xxx.100.13  host02.example.com  host02
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The following is a list of requirements for network configuration:

- Each node must have at least two network interface cards (NICs): one for the public network and one for the private network.
- The network interface names must be the same across all nodes in the cluster.
- On the public network, each NIC must support TCP/IP.
- On the private network, each NIC must support User Datagram Protocol (UDP).
- If domain name servers (DNS) are being used, the public IP addresses should be registered.
- Ensure that each node is properly identified using the `hostname` and `ifconfig` utilities.
- If the time stamps among the nodes differ significantly, node evictions and reboots can occur. Network Time Protocol (`ntpd`) can be used to synchronize time stamps between cluster nodes. If NTP is not configured, Oracle Clusterware installs a cluster time daemon, `csstd`, in observer mode.

You can configure IP addresses with one of the following options:

- Oracle Grid Naming Service (GNS) using one static address defined during installation, with dynamically allocated addresses for all other IP addresses, obtained from your organization's Dynamic Host Configuration Protocol (DHCP) server, and resolved using a multicast domain name server configured within the cluster
- Static addresses that network administrators manually assign on a network domain name server (DNS) for each node

IP Address Requirements with GNS

- If GNS is used, name resolution requests to the cluster are delegated to the GNS, which is listening on the GNS VIP.
- The GNS VIP address is defined in the DNS domain before installation.
- The DNS must be configured to delegate resolution requests for cluster names to the GNS.
- Before installation, the DNS administrator must establish DNS Lookup to direct the DNS resolution of a subdomain to the cluster.
- A DHCP service on the public network is required that allows the cluster to dynamically allocate the VIP addresses as required by the cluster.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

IP Address Requirements for Static Configuration

- If you do not enable GNS or enable GNS without subdomain delegation, then the public and virtual IP addresses for each node must be static IP addresses.
- The addresses must be configured before Clusterware installation but must not be currently in use.
- Public and virtual IP addresses must be on the same subnet.
- The cluster must have the following addresses configured:
 - A public IP address for each node
 - A virtual IP address for each node
 - A private IP address for each node
 - A Single-Client Access Name (SCAN) for the cluster



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

If you do not enable GNS or enable GNS without subdomain delegation, the public and virtual IP addresses for each node must be static IP addresses, configured before installation for each node but not currently in use. Public and virtual IP addresses must be on the same subnet.

Oracle Clusterware manages private IP addresses in the private subnet on interfaces you identify as private during the installation process. The cluster must have the following addresses configured:

- A public IP address for each node, with the following characteristics:
 - Static IP address
 - Configured before installation for each node, and resolvable to that node before installation
 - On the same subnet as all other public IP, VIP, and SCAN addresses
- A virtual IP address for each node, with the following characteristics:
 - Static IP address
 - Configured before installation for each node, but not currently in use
 - On the same subnet as all other public IP addresses, VIP addresses, and SCAN addresses

- A Single-Client Access Name (SCAN) for the cluster, with the following characteristics:
 - Three Static IP addresses configured on the domain name server (DNS) before installation so that the three IP addresses are associated with the name provided as the SCAN, and all three addresses are returned in random order by the DNS to the requestor
 - Configured before installation in the DNS to resolve to addresses that are not currently in use
 - Given a name that does not begin with a numeral
 - On the same subnet as all other public IP addresses, VIP addresses, and SCAN addresses
 - Conforms with the RFC 952 standard, which allows alphanumeric characters and hyphens (“-”), but does not allow underscores (“_”)
- A private IP address for each node, with the following characteristics:
 - Static IP address
 - Configured before installation, but on a separate private network, with its own subnet, that is not resolvable except by other cluster member nodes

The SCAN is a name used to provide service access for clients to the cluster. Because the SCAN is associated with the cluster as a whole, rather than to a particular node, the SCAN makes it possible to add nodes to or remove nodes from the cluster without needing to reconfigure clients. It also adds location independence for the databases, so that client configuration does not have to depend on which nodes are running a particular database. Clients can continue to access the cluster in the same way as with previous releases, but Oracle recommends that clients accessing the cluster use the SCAN.

Broadcast and Multicast Requirements

- Broadcast communications (ARP and UDP) must work properly across all the public and private interfaces.
- The broadcast must work across any configured VLANs as used by the public or private interfaces.
- The Oracle mDNS daemon uses multicasting on all interfaces to communicate with other nodes in the cluster.
- With Oracle Grid Infrastructure 11.2.0.2 and later releases, multicasting is required on the private interconnect.
- Multicasting must be enabled for the cluster:
 - Across the broadcast domain as defined for the private interconnect
 - On the IP address subnet ranges 224.0.0.0/24 and 230.0.1.0/24



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Broadcast communications (ARP and UDP) must work properly across all the public and private interfaces configured for use by Oracle Grid Infrastructure. The broadcast must work across any configured VLANs as used by the public or private interfaces.

With Oracle Grid Infrastructure on each cluster member node, the Oracle mDNS daemon uses multicasting on all interfaces to communicate with other nodes in the cluster. Multicasting is required on the private interconnect. For this reason, at a minimum, you must enable multicasting for the cluster:

- Across the broadcast domain as defined for the private interconnect
- On the IP address subnet ranges 224.0.0.0/24 and 230.0.1.0/24 (Classless Inter-Domain Routing – CIDR)

You do not need to enable multicast communications across routers.

Private Interconnect Network Requirements

- For clusters using single interfaces for private networks:
 - Each node's private interface must be on the same subnet
 - The subnet must connect to every node of the cluster
- For clusters using Redundant Interconnect Usage:
 - Each private interface should be on a different subnet
 - Each cluster member node must have an interface on each private interconnect subnet
 - These subnets must connect to every node of the cluster
- For the private network, the endpoints of all interconnect interfaces must be completely reachable on the network.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

For clusters using single interfaces for private networks, each node's private interface for interconnects must be on the same subnet, and that subnet must connect to every node of the cluster. For example, if the private interfaces have a subnet mask of 255.255.255.0, then your private network is in the range 192.168.0.0 – 192.168.0.255, and your private addresses must be in the range of 192.168.0.[0–255]. If the private interfaces have a subnet mask of 255.255.0.0, your private addresses can be in the range of 192.168.[0–255].[0–255]. For clusters incorporating Redundant Interconnect Usage, each private interface should be on a different subnet. However, each cluster member node must have an interface on each private interconnect subnet, and these subnets must connect to every node of the cluster. For example, you can have private networks on subnets 192.168.0 and 10.0.0, but each cluster member node must have an interface connected to the 192.168.0 and 10.0.0 subnets.

For the private network, the endpoints of all designated interconnect interfaces must be completely reachable on the network. There should be no node that is not connected to every private network interface. You can test if an interconnect interface is reachable using ping.

Interconnect NIC Guidelines

Optimal interconnect NIC settings can vary depending on the driver used. Consider the following guidelines:

- Configure the interconnect NIC on the fastest PCI bus.
- Ensure that NIC names and slots are identical on all nodes.
- Define flow control: receive=on, transmit=off.
- Define full bit rate supported by NIC.
- Define full duplex autonegotiate.
- Ensure compatible switch settings:
 - If 802.3ad is used on NIC, it must be used and supported on the switch.
 - The Maximum Transmission Unit (MTU) should be the same between NIC and the switch.
- Driver settings can change between software releases.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Failure to correctly configure the network interface cards and switches used for the interconnect results in severe performance degradation and possible node evictions or node fencing. If there is a choice between bus standards such as PCI and PCI express, configure the interconnect NIC on the fastest PCI bus. It is a requirement for the NIC devices and switch to autonegotiate to achieve the highest supported bit rate possible. Flow control should be turned on for receive. Cases have occurred where this setting has been altered between driver software updates and changes. Depending on the mode of link aggregation used, specialized support may be needed at the switch. Synchronization between the switch settings and network interface cards is very important.

For Oracle database Real Application Clusters (RAC), the interconnect will be used to transport database block images. An Oracle database block can be sized up to 32 KB, whereas a typical interconnect communication message averages around 200 bytes.

A misconfigured or faulty interconnect can lead to a variety of problems such as:

- Dropped packets and fragments
- Buffer overflows
- Packet reassembly failures or timeouts
- General Tx/Rx errors

Private Interconnect Redundant Network Requirements

- If you use Oracle Clusterware Redundant Interconnect, then you can use either IPv4 network or IPv6 network for the interfaces.
- When you define multiple interfaces, Clusterware creates from one to four highly available IP (HAIP) addresses.
- The installer enables Redundant Interconnect Usage to provide a high availability private network.
- Grid Infrastructure uses all of the HAIP addresses for the private network, load-balancing private interconnect traffic.
- If a private interconnect interface fails, then Clusterware transparently moves the HAIP address to one of the remaining functional interfaces.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

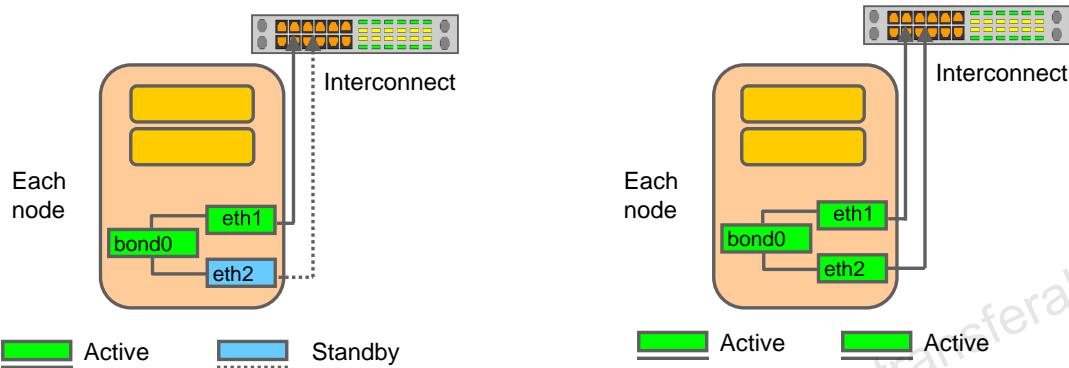
With Redundant Interconnect Usage, you can identify multiple interfaces to use for the cluster private network, without the need of using bonding or other technologies. This functionality is available starting with Oracle Database 11g Release 2 (11.2.0.2).

If you use the Oracle Clusterware Redundant Interconnect feature, you can use either IPv4 or IPv6 addresses for the interfaces. However, you cannot mix IPv4 and IPv6 addresses for any private network interfaces. When you define multiple interfaces, Oracle Clusterware creates from one to four highly available IP (HAIP) addresses. Oracle RAC and Oracle Automatic Storage Management (Oracle ASM) instances use these interface addresses to ensure highly available, load-balanced interface communication between nodes. The installer enables Redundant Interconnect Usage to provide a high-availability private network.

By default, Oracle Grid Infrastructure software uses all of the HAIP addresses for private network communication, providing load-balancing across the set of interfaces you identify for the private network. If a private interconnect interface fails or becomes noncommunicative, Oracle Clusterware transparently moves the corresponding HAIP address to one of the remaining functional interfaces.

Interconnect Link Aggregation: Single Switch

- Link aggregation can be used to increase redundancy for higher availability with an Active/Standby configuration.
- Link aggregation can be used to increase bandwidth for performance with an Active/Active configuration.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Interconnect link aggregation involves bonding two or more physical network interface cards into a single logical “bonded” interface. The behavior of the bonded interfaces depends on the settings, modes, and drivers used to accomplish the aggregation.

One strategy often used for highly available configurations is the Active/Standby aggregation, sometimes known as Active/Backup or Active/Passive aggregation. Generally, only one of the network interface cards carries traffic, and the other is available for failover. An example of an Active/Backup setup on Linux as reported with the `ifconfig` command is as follows:

```
bond0 Link encap:Ethernet HWaddr 00:C0:F0:1F:37:B4
          inet addr:XXX.XXX.XXX.YYY Bcast:XXX.XXX.XXX.255 Mask:255.255.252.0
              UP BROADCAST RUNNING MASTER MULTICAST MTU:1500 Metric:1
              RX packets:7224794 errors:0 dropped:0 overruns:0 frame:0
              TX packets:3286647 errors:1 dropped:0 overruns:1 carrier:0
              collisions:0 txqueuelen:0
```

```
eth1    Link encap:Ethernet  HWaddr 00:C0:F0:1F:37:B4
        inet addr:XXX.XXX.XXX.YYY  Bcast:XXX.XXX.XXX.255
Mask:255.255.252.0
              UP BROADCAST RUNNING NOARP SLAVE MULTICAST  MTU:1500  Metric:1
              RX packets:3573025 errors:0 dropped:0 overruns:0 frame:0
              TX packets:1643167 errors:1 dropped:0 overruns:1 carrier:0
              collisions:0 txqueuelen:100
              Interrupt:10 Base address:0x1080

eth2    Link encap:Ethernet  HWaddr 00:C0:F0:1F:37:B4
        inet addr:XXX.XXX.XXX.YYY  Bcast:XXX.XXX.XXX.255
Mask:255.255.252.0
              UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
              RX packets:3651769 errors:0 dropped:0 overruns:0 frame:0
              TX packets:1643480 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:100
              Interrupt:9 Base address:0x1400
```

The `eth1` and `eth2` devices are physical network interface cards. The `bond0` device is a “virtual” network interface card. Notice that the logical `bond0` interface is listed as the **MASTER**, and the other two interfaces are listed as **SLAVE**. The interface device without the NOARP (`eth2`) is the current active **SLAVE**. Also notice that all three interfaces report the same layer-2 or Media Access Control (MAC) address and have IP addresses. Traffic statistics exist on all Network Interface Card (NIC) devices in this sample output because of extended up time and failures in the past.

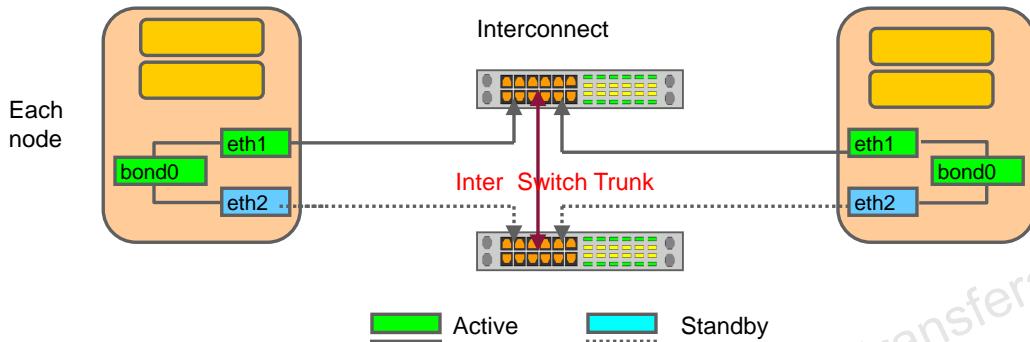
Note: During the installation of the Oracle Clusterware product, the `bond0` interface will be supplied as the value to the prompts for the interconnect interface to be used.

Another common strategy for link aggregation involves Active/Active configurations following the IEEE 802.3ad standards. This arrangement involves simultaneous use of both the bonded physical network interface cards in parallel to achieve a higher bandwidth beyond the limit of any one single network card. It is very important that if 802.3ad is used at the NIC layer, the switch must also support and be configured for 802.3ad. Misconfiguration results in poor performance and interface resets or “port flapping.” An alternative is to consider a single network interface card with a higher bandwidth, such as 10 Gb Ethernet instead of 1Gb Ethernet. InfiniBand can also be used for the interconnect.

Link aggregation is sometimes known as “NIC teaming,” “NIC bonding,” “port trunking,” “EtherChannel,” “Multi-Link Trunking (MLT),” “Network Fault Tolerance (NFT),” and “link aggregate group (LAG).” Link aggregation is usually limited to a single switch. Multiswitch solutions include “Split Multi-Link Trunking (SMLT),” “Distributed Split Multi-Link Trunking (DSMLT),” and “Routed Split Multi-Link Trunking (RSMLT).”

Interconnect Link Aggregation: Multiswitch

- Redundant switches connected with an Inter-Switch Trunk can be used for an enhanced highly available design.
- This is the best practice configuration for the interconnect.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

With the single-switch solutions presented in the previous slide, a failure at the switch level would bring down the entire interconnect. A better highly available (HA) design would be to implement a redundant switch strategy as illustrated in the slide, with an Inter-Switch Trunk connecting the switches. This is the best practice design for the Oracle Clusterware interconnect. Only Active/Standby mode is supported in this configuration.

Some of the standard aggregation solutions include:

- Cisco EtherChannel based on 802.3ad
- AIX EtherChannel
- HPUX Auto Port Aggregation
- Sun Trunking, IPMP, GLD
- Linux Bonding (only certain modes)
- Windows NIC teaming

A virtual LAN (VLAN) is supported in a shared switch environment for the interconnect. The interconnect should be a dedicated, nonroutable subnet mapped to a single, dedicated, nonshared VLAN.

Additional Interconnect Guidelines

UDP socket buffer (rx):

- Default settings are adequate for the majority of customers.
- It may be necessary to increase the allocated buffer size when the:
 - MTU size has been increased
 - netstat command reports errors
 - ifconfig command reports dropped packets or overflow

Jumbo frames:

- Are not an Institute of Electrical and Electronics Engineers (IEEE) standard
- Are useful for network-attached storage (NAS)/iSCSI storage
- Have network device interoperability concerns
- Need to be configured with care and tested rigorously



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The maximum UDP socket receive buffer size varies according to the operating system. The upper limit may be as small as 128 KB or as large as 1 GB. In most cases, the default settings are adequate for the majority of customers. This is one of the first settings to consider if you are receiving lost blocks. Consult the My Oracle Support (formerly, MetaLink) website for best-practice settings for your platform. Three significant conditions that indicate when it may be necessary to change the UDP socket receive buffer size are when the MTU size has been increased, when excessive fragmentation and/or reassembly of packets is observed, and if dropped packets or overflows are observed.

Jumbo frames are not a requirement for Oracle Clusterware and not configured by default. The use of jumbo frames is supported; however, special care must be taken because this is not an IEEE standard and there are significant variances among network devices and switches especially from different manufacturers. The typical frame size for jumbo frames is 9 KB, but again, this can vary. It is necessary that all devices in the communication path be set to the same value.

Note: For Oracle Clusterware, the Maximum Transmission Unit (MTU) needs to be the same on all nodes. If it is not set to the same value, an error message will be sent to the Clusterware alert logs.

Cluster Time Synchronization

- Oracle Clusterware requires the same time zone environment variable setting on all cluster nodes.
- There are two options for time synchronization:
 - An operating system–configured network time protocol (NTP)
 - Oracle Cluster Time Synchronization Service (CTSS)
- CTSS is designed for cluster servers that are unable to access NTP services.
- If NTP is used, then the `octssd` daemon starts up in observer mode.
- If NTP is not installed or is deactivated, CTSS is used and the `octssd` process is started in active mode.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware requires the same time zone environment variable setting on all cluster nodes. During installation, the installation process picks up the time zone environment variable setting of the Grid installation owner on the node where OUI runs, and uses that time zone value on all nodes as the default TZ environment variable setting for all processes managed by Oracle Clusterware. The time zone default is used for databases, Oracle ASM, and any other managed processes. You have two options for time synchronization:

- An operating system configured network time protocol (NTP)
- Oracle Cluster Time Synchronization Service

Oracle Cluster Time Synchronization Service is designed for organizations whose cluster servers are unable to access NTP services. If you use NTP, then the Oracle Cluster Time Synchronization daemon (`ctssd`) starts up in observer mode. If you do not have NTP daemons, then `octssd` starts up in active mode and synchronizes time among cluster members without contacting an external time server.

If you have NTP daemons on your server but you cannot configure them to synchronize time with a time server, and you want to use Cluster Time Synchronization Service to provide synchronization service in the cluster, you should deactivate and uninstall the NTP.

To deactivate the NTP service, you must stop the existing `ntpd` service, disable it from the initialization sequences, remove the `ntp.conf` and `ntpd.pid` files.

To complete these steps on Oracle Linux, and Asianux systems, run the following commands as the root user:

```
# /sbin/service ntpd stop  
# chkconfig ntpd off  
# mv /etc/ntp.conf /etc/ntp.conf.org  
# rm /var/run/ntp.pid
```

When the installer finds that the NTP protocol is not active, the Cluster Time Synchronization Service is installed in active mode and synchronizes the time across the nodes. If NTP is found configured, the Cluster Time Synchronization Service is started in observer mode, and no active time synchronization is performed by Oracle Clusterware within the cluster.

To confirm that ctssd is active after installation, enter the following command as the Grid installation owner:

```
$ crsctl check ctssd
```

For NTP, the maximum slew rate possible is limited to .5 ms/s as a consequence of the principles on which the NTP protocol and algorithm design are based. As a result, the local clock can take a long time to converge to an acceptable offset, about 2,000s for each second the clock is outside the acceptable range. As a result, an adjustment as much as five minutes (300s) will take almost seven days to complete. During this interval, the local clock will not be consistent with other network clocks and the system cannot be used for distributed applications that require correctly synchronized network time. As a result of this behavior, after the clock has been set, it very rarely strays more than 128 ms, even under extreme cases of network path congestion and jitter. Sometimes, in particular when `ntpd` is first started, the error might exceed 128 ms. This may on occasion cause the clock to be stepped backward if the local clock time is more than 128 ms in the future relative to the server. In some applications, this behavior may be unacceptable. If the `-x` option is included on the command line, the clock will never be stepped and only slew corrections will be used. In practice, this reduces the false alarm rate where the clock is stepped in error to a diminishingly low incidence.

Software Requirements (Kernel)

- Supported Linux distributions and kernel requirements:

| Linux Distribution | Requirements |
|---------------------------------|--|
| Oracle Linux (OL) | Oracle Linux 7 with UEK 3: 3.8.13-35.3.1.el7uek.x86_64 or later Oracle Linux 7.2 with UEK 4: 4.1.12-32.2.3.el7uek.x86_64 or later Oracle Linux 7 with Red Hat Compatible kernel: 3.10.0-123.el7.x86_64 or later Oracle Linux 6.4 with UEK 2: 2.6.39-400.211.1.el6uek.x86_64 or later Oracle Linux 6.6 with UEL 3: 3.8.13-44.1.1.el6uek.x86_64 or later Oracle Linux 6.8 with UEL 4: 4.1.12-37.6.2.el6uek.x86_64 or later Oracle Linux 6.4 with Red Hat Compatible kernel: 2.6.32-358.el6.x86_64 or later |
| Red Hat Enterprise Linux (RHEL) | Red Hat Enterprise Linux 7: 3.10.0-123.el7.x86_64 or later Red Hat Enterprise Linux 6.4: 2.6.32-358.el6.x86_64 or later |
| SUSE Enterprise Linux | SUSE Linux Enterprise Server 12 SP1: 3.12.49-11.1 or later |



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Software Requirements: Packages

- Oracle Grid Infrastructure and Oracle RAC Linux x86 64-bit RPM requirements for OL 6 and RHEL 6

```
binutils-2.20.51.0.2-5.36.el6 (x86_64)
compat-libcap1-1.10-1 (x86_64)
compat-libstdc++-33-3.2.3-69.el6 (x86_64)
compat-libstdc++-33-3.2.3-69.el6 (i686)
e2fsprogs-1.41.12-14.el6 (x86_64)
e2fsprogs-libs-1.41.12-14.el6 (x86_64)
glibc-2.12-1.7.el6 (i686)
glibc-2.12-1.7.el6 (x86_64)
glibc-devel-2.12-1.7.el6 (x86_64)
glibc-devel-2.12-1.7.el6 (i686)
ksh
libgcc-4.4.4-13.el6 (i686)
libgcc-4.4.4-13.el6 (x86_64)
libstdc++-4.4.4-13.el6 (x86_64)
libstdc++-4.4.4-13.el6 (i686)
libstdc++-devel-4.4.4-13.el6 (x86_64)
libstdc++-devel-4.4.4-13.el6 (i686)
libaio-0.3.107-10.el6 (x86_64)
libaio-0.3.107-10.el6 (i686)
libaio-devel-0.3.107-10.el6 (x86_64)

libaio-devel-0.3.107-10.el6 (i686)
libXtst-1.0.99.2 (x86_64)
libXtst-1.0.99.2 (i686)
libX11-1.5.0-4.el6 (i686)
libX11-1.5.0-4.el6 (x86_64)
libXau-1.0.6-4.el6 (i686)
libXau-1.0.6-4.el6 (x86_64)
libxcb-1.8.1-1.el6 (i686)
libxcb-1.8.1-1.el6 (x86_64)
libXi-1.3 (x86_64)
libXi-1.3 (i686)
make-3.81-19.el6
net-tools-1.60-110.el6_2.x86_64 (for Oracle RAC and
Oracle
Clusterware)
nfs-utils-1.2.3-15.0.1 (for Oracle ACFS)
sysstat-9.0.4-11.el6 (x86_64)
smartmontools-5.43-1.el6.x86_64
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

For Linux x86 64-bit systems running OL 6, or RHEL 6, the packages listed in the slide (or later versions) must be installed.

Software Requirements: Packages

- Oracle Grid Infrastructure and Oracle RAC Linux x86 64-bit RPM requirements for OL 7 and RHEL 7

```
compat-libcap1-1.10-3.el7 (x86_64)
compat-libstdc++-33-3.2.3-71.el7 (i686)
compat-libstdc++-33-3.2.3-71.el7 (x86_64)
glibc-2.17-36.el7 (i686)
glibc-2.17-36.el7 (x86_64)
glibc-devel-2.17-36.el7 (i686)
glibc-devel-2.17-36.el7 (x86_64)
ksh
libaio-0.3.109-9.el7 (i686)
libaio-0.3.109-9.el7 (x86_64)
libaio-devel-0.3.109-9.el7 (i686)
libaio-devel-0.3.109-9.el7 (x86_64)
libX11-1.6.0-2.1.el7 (i686)
libX11-1.6.0-2.1.el7 (x86_64)
libXau-1.0.8-2.1.el7 (i686)
libXau-1.0.8-2.1.el7 (x86_64)
libXi-1.7.2-1.el7 (i686)
libXi-1.7.2-1.el7 (x86_64)
libXtst-1.2.2-1.el7 (i686)

libgcc-4.8.2-3.el7 (i686)
libgcc-4.8.2-3.el7 (x86_64)
libstdc++-4.8.2-3.el7 (i686)
libstdc++-4.8.2-3.el7 (x86_64)
libstdc++-devel-4.8.2-3.el7 (i686)
libstdc++-devel-4.8.2-3.el7 (x86_64)
libxcb-1.9-5.el7 (i686)
libxcb-1.9-5.el7 (x86_64)
make-3.82-19.el7 (x86_64)
nfs-utils-1.3.0-0.21.el7.x86_64 (for Oracle ACFS)
net-tools-2.0-0.17.20131004git.el7 (x86_64)
(for Oracle RAC and Oracle Clusterware)
smartmontools-6.2-4.el7 (x86_64)
sysstat-10.1.5-1.el7 (x86_64)
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

For Linux x86 64-bit systems running OL 7, or RHEL 7, the packages listed in the slide (or later versions) must be installed.

Software Requirements: Packages

- Oracle Grid Infrastructure and Oracle RAC Linux x86 64-bit RPM requirements for SUSE Linux Enterprise Server 12

```
binutils-2.24-2.165.x86_64
gcc-c++-32bit-4.8-6.189.x86_64
gcc-c++-4.8-6.189.x86_64
gcc48-c++-4.8.3+r212056-6.3.x86_64
gcc-32bit-4.8-6.189.x86_64
gcc-4.8-6.189.x86_64
gcc-info-4.8-6.189.x86_64
gcc-locale-4.8-6.189.x86_64
gcc48-32bit-4.8.3+r212056-6.3.x86_64
gcc48-4.8.3+r212056-6.3.x86_64
gcc48-info-4.8.3+r212056-6.3.noarch
gcc48-locale-4.8.3+r212056-6.3.x86_64
glibc-2.19-17.72.x86_64
glibc-devel-2.19-17.72.x86_64
libaio-devel-0.3.109-17.15.x86_64
libaio1-0.3.109-17.15.x86_64
libaio1-32bit-0.3.109-17.15.x86_64
libgfortran3-4.8.3+r212056-6.3.x86_64
libX11-6-1.6.2-4.12.x86_64
libXau6-1.0.8-4.58.x86_64
libXau6-32bit-1.0.8-4.58.x86_64
libXtst6-1.2.2-3.60.x86_64
libXtst6-32bit-1.2.1-2.4.1.x86_64
libcap-ng-utils-0.7.3-4.125.x86_64
libcap-ng0-0.7.3-4.125.x86_64
libcap-ng0-32bit-0.7.3-4.125.x86_64
libcap-progs-2.22-11.709.x86_64
libcap1-1.10-59.61.x86_64
libcap1-32bit-1.10-59.61.x86_64
libcap2-2.22-11.709.x86_64
libcap2-32bit-2.22-11.709.x86_64
libgcc_s1-32bit-4.8.3+r212056-6.3.x86_64
libgcc_s1-4.8.3+r212056-6.3.x86_64
libpcap1-1.5.3-2.18.x86_64
libstdc++-6-32bit-4.8.3+r212056-6.3.x86_64
libstdc++-6-4.8.3+r212056-6.3.x86_64
make-4.0-2.107.x86_64
mksh-50-2.13.x86_64
net-tools-1.60-764.185.x86_64 (for Oracle RAC
and Oracle Clusterware)
nfs-kernel-server-1.3.0-6.9.x86_64 (for Oracle
ACFS)
smartmontools-6.2-4.33.x86_64
sysstat-8.1.5-7.32.1.x86_64
xorg-x11-libs-7.6-45.14
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

For Linux x86 64-bit systems running SUSE Linux Enterprise Server 12, the packages listed in the slide (or later versions) must be installed.

Oracle Linux with the Unbreakable Enterprise Kernel

- Oracle's Unbreakable Enterprise Kernel delivers the latest innovations to customers running OL or RHEL.
- The Unbreakable Enterprise Kernel is included and enabled by default starting with Oracle Linux 5 Update 6.
- The Unbreakable Enterprise Kernel:
 - Is based on a recent stable mainline development Linux kernel
 - Includes optimizations developed in collaboration with Oracle Database, Oracle middleware, and Oracle hardware engineering teams
 - Ensures stability and optimal performance for the most demanding enterprise workloads
- A RHEL-compatible kernel is provided for organizations that need strict RHEL compatibility.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle's Unbreakable Enterprise Kernel delivers the latest innovations from upstream development to customers who run Red Hat Enterprise Linux (RHEL) 5 or Oracle Linux 5 in the data center.

Oracle Corporation highly recommends deploying the Oracle Unbreakable Enterprise Kernel in your Linux environment, especially if you run enterprise applications; however, its usage is optional.

If you require strict RHEL kernel compatibility, Oracle Linux also includes a kernel compatible with the RHEL Linux kernel, compiled directly from the RHEL source code. You can obtain more information about the Oracle Unbreakable Enterprise Kernel for Linux at the following URL:

<http://www.oracle.com/us/technologies/linux>

The Oracle Unbreakable Enterprise Kernel for Linux is the standard kernel used with Oracle products. The build and QA systems for Oracle Database and other Oracle products use the Oracle Unbreakable Enterprise Kernel for Linux exclusively.

The Oracle Unbreakable Enterprise Kernel for Linux is also the kernel used in Oracle Exadata and Oracle Exalogic systems. Oracle Unbreakable Enterprise Kernel for Linux is used in all benchmark tests on Linux in which Oracle participates, as well as in the Oracle RDBMS preinstall RPM program for x86-64.

Zero-Downtime Kernel Updates with Ksplice

- Ksplice updates the Linux kernel while it is running.
 - Ksplice is part of the Oracle Linux distribution.
- To configure Ksplice Repository and perform a kernel update:
 1. Check for your kernel distribution at the following URL:
<http://www.ksplice.com/uptrack/supported-kernels>
 2. Download the Ksplice Uptrack repository RPM package at:
<https://www.ksplice.com/yum/uptrack/ol/ksplice-uptrack-release.noarch.rpm>
 3. Install Ksplice and Upstart RPMs with commands:

```
# rpm -i ksplice-uptrack-release.noarch.rpm
# yum -y install uptrack
```
 4. Execute the uptrack-upgrade command as the `root` user:

```
# uptrack-update -y
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Ksplice, which is part of Oracle Linux, updates the Linux operating system (OS) kernel, while it is running, without requiring restarts or any interruption. Ksplice is available only with Oracle Linux. You can use Ksplice if you have Premier support subscription and an access key, which is available on ULN. For more information about Ksplice, see <http://www.ksplice.com>. To configure Ksplice Repository for Oracle Linux and perform a zero-downtime upgrade of your kernel, perform the following steps:

1. Check for your kernel distribution at the following URL:
<http://www.ksplice.com/uptrack/supported-kernels#>
2. Ensure that you have access to the Internet on the server where you want to use Ksplice.
3. Download the Ksplice Uptrack repository RPM package:
<https://www.ksplice.com/yum/uptrack/ol/ksplice-uptrack-release.noarch.rpm>
4. As the `root` user, run the following commands:

```
# rpm -i ksplice-uptrack-release.noarch.rpm
# yum -y install uptrack
```
5. Open `/etc/uptrack/uptrack.conf` with a text editor, enter your premium support access key, and save the file. You must use the same access key for all of your systems.
6. Run the following command to carry out a zero down time update of your kernel:

```
# uptrack-upgrade -y
```

Note: Currently, Ksplice is available only with Premium Support.

Oracle Preinstallation RPM

- Use the Oracle Database Server 12c Release 2 Preinstallation RPM to complete most configuration tasks for OL and RHEL.
- When it is installed, the Oracle Preinstallation RPM:
 - Automatically downloads and installs any additional RPMs needed for Grid Infrastructure and Oracle RAC
 - Creates an `oracle` user, and creates the `oraInventory` (`oinstall`) and OSDBA (`dba`) groups
 - Sets `sysctl.conf` settings, system startup parameters, and driver parameters
 - Sets hard and soft resource limits
 - Sets other recommended parameters, depending on your kernel version
- For Oracle Linux Release 5.2 and higher, the Oracle Preinstallation RPM is included on the install media.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

If your Linux distribution is Oracle Linux, or Red Hat Enterprise Linux, and you are an Oracle Linux customer, then you can complete most Preinstallation configuration tasks by using the Oracle Database Server 12cR2 Pre-Installation RPM, available from the Oracle Linux Network, or available on the Oracle Linux DVDs. Using the Oracle Pre-Installation RPM is not required, but Oracle recommends you use it to save time in setting up your cluster servers.

When it is installed, the Oracle Preinstallation RPM does the following:

- Automatically downloads and installs any additional RPM packages needed for installing Oracle Grid Infrastructure and Oracle Database, and resolves any dependencies
- Creates an `oracle` user, and creates the `oraInventory` (`oinstall`) and OSDBA (`dba`) groups for that user
- Sets `sysctl.conf` settings, system startup parameters, and driver parameters to values based on recommendations from the Oracle RDBMS Preinstallation program
- Sets hard and soft resource limits
- Sets other recommended parameters, depending on your kernel version

If you are using Oracle Linux 5.2 and higher, then the Oracle Pre-Install RPM is included on the install media. The Oracle Preinstallation RPM does not install OpenSSH, which is required for Oracle Grid Infrastructure installation. If you perform a minimal Linux installation and install the Oracle Preinstallation RPM for your release, then you must also install the OpenSSH client manually. Using RSH is no longer supported.

Installing the cvuqdisk RPM for Linux

- If you do not use an Oracle Preinstallation RPM, then you must install the `cvuqdisk` RPM.
- Without `cvuqdisk`, Cluster Verification Utility cannot discover shared disks.
- To install the `cvuqdisk` RPM:
 1. Set the environment variable `CVUQDISK_GRP` to point to the group that will own `cvuqdisk`:

```
# export CVUQDISK_GRP=oinstall
```

2. Install the `cvuqdisk` RPM from the directory where it was saved:

```
# rpm -iv cvuqdisk-1.0.10-1.rpm
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

If you do not use an Oracle Pre-Install RPM, then you must install the `cvuqdisk` RPM. Without `cvuqdisk`, Cluster Verification Utility cannot discover shared disks, and you receive the error message “Package `cvuqdisk` not installed” when you run Cluster Verification Utility. Use the `cvuqdisk` rpm for your hardware (for example, `x86_64`).

To install the `cvuqdisk` RPM, complete the following procedure:

1. Locate the `cvuqdisk` RPM package, which is in the directory `rpm` on the Oracle Grid Infrastructure installation media.
2. Copy the `cvuqdisk` package to each node on the cluster.
3. Use the following command to find if you have an existing version of the `cvuqdisk` package:

```
# rpm -qi cvuqdisk
```

If you have an existing version, uninstall the existing version:

```
# rpm -e cvuqdisk
```

4. Set the environment variable `CVUQDISK_GRP` to point to the group that will own `cvuqdisk`, typically `oinstall`. For example:

```
# CVUQDISK_GRP=oinstall; export CVUQDISK_GRP
```

5. In the directory where you have saved the `cvuqdisk` rpm, use the `rpm -iv package` command to install the `cvuqdisk` package. For example:

```
# rpm -iv cvuqdisk-1.0.10-1.rpm
```

Creating Groups and Users

- Create an Oracle Software inventory group on each node.
- Group ID must be consistent on each node.

```
# groupadd -g 54321 oinstall
```

- Create the Oracle Software owner on each node.
- User ID must be consistent on each node, and the inventory group must be the primary group.
- Most Oracle products are usually owned by the same user, typically called `oracle`, but each product can be owned by a different user.

```
# useradd -u 54321 -g oinstall oracle  
or  
# useradd -u 54322 -g oinstall grid
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

An operating system group needs to be created that will be associated with Oracle Central Inventory (`oraInventory`). `oraInventory` contains a registry of the Oracle home directories from all Oracle products installed on the server. It is designed to be shared among many products. It also contains installation log files and trace files from the installation programs. The suggested name for the operating system group is `oinstall`. In a cluster installation, it is very important that the group ID be the same on all nodes of the cluster.

An operating system user needs to be created to own the Oracle Clusterware installation. Traditionally, all Oracle products installed on the same machine such as clusterware, databases, disk management, and enterprise management tools are owned by the same user called `oracle`. It is possible for each product to be created under a different operating system account. This may be desired if different job responsibilities are used to manage different components. It is very important that the user ID be the same on all nodes of the cluster.

It is required that the Oracle Clusterware software owner and the Oracle Database software owner have the same primary group as the Oracle Inventory group.

Note: If this installation of Oracle Clusterware contains a database and other Oracle products, consider creating the following groups: `dba` and `oper`.

Creating Groups, Users, and Paths

1

Create Groups:

```
# /usr/sbin/groupadd -g 54321 oinstall
# /usr/sbin/groupadd -g 54322 dba
# /usr/sbin/groupadd -g 54329 oper
# /usr/sbin/groupadd -g 54324 asmdba
# /usr/sbin/groupadd -g 54327 asmadmin
# /usr/sbin/groupadd -g 54328 asmoper
```

2

Create Users

```
# /usr/sbin/useradd -u 54321 -g oinstall -G asmdba,dba,oper oracle
# /usr/sbin/useradd -u 54322 -g oinstall -G asmdba,asmadmin,asmoper grid
```

3

Create Directories

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



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The following is an example of how to create the Oracle Inventory group (`oinstall`), OSDBA, OSASM, and OSOPER for ASM groups where they are the same group (`dba`), and how to create the Grid Infrastructure software owner (`grid`) and one Oracle Database owner (`oracle`) with correct group memberships. This example shows how to configure an Oracle base path compliant with the Optimal Flexible Architecture structure with correct permissions:

1. Log in as root. Enter commands similar to the example in the slide to create the `oinstall`, `asmadmin`, and `asmdba` groups, and if required, the `asmoper`, `dba`, and `oper` groups. Use the `-g` option to specify the correct group ID for each group.
2. To create the Grid Infrastructure user, enter a command similar to the following (in this example):

```
# /usr/sbin/useradd -u 1100 -g oinstall -G asmdba,dba,asmoper oracle
```

If a separate grid owner is needed:

```
# /usr/sbin/useradd -u 1100 -g oinstall -G asmdba,dba,oper oracle
# /usr/sbin/useradd -u 1101 -g oinstall -G asmdba,asmadmin,asmoper grid
```

3. Create base directories for Oracle Grid Infrastructure and Database software:

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

Shell Settings for the Grid Infrastructure User

- Add the following lines to the `/etc/security/limits.conf` file on each node:

```
grid soft nproc 2047
grid hard nproc 16384
grid soft nofile 1024
grid hard nofile 65536
grid soft stack 10240
grid hard stack 32768
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
oracle soft stack 10240
oracle hard stack 32768
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Determining Root Script Execution Plan

- During Grid Infrastructure installation, you must run scripts with root privileges to complete some configuration tasks.
- With Oracle Grid Infrastructure 12c, you can:
 - Run scripts manually as `root`
 - Delegate to the installer the privilege to run configuration steps as root automatically
- There are two choices to run root scripts automatically:
 - Provide the password to OUI as you are providing other configuration information.
 - Use `sudo` to grant privileges for root scripts to non-root users, and then provide the privileged user and password to OUI.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.



Quiz

Which of the following statements are not true?

- a. At least 4 GB of physical memory is needed.
- b. A minimum of 3 GB of swap space is required.
- c. The local /tmp directory should have at least 1 GB free.
- d. At least 8 GB of local storage for the Grid Infrastructure software is needed on each node.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to perform preinstallation tasks for Grid Infrastructure



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Practice 4: Overview

This practice covers the following topic:

- Performing the required Preinstallation Tasks



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Grid Infrastructure Installation



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Perform an image-based Grid Infrastructure installation
- Verify the installation
- Configure Automatic Storage Management (ASM) disk groups

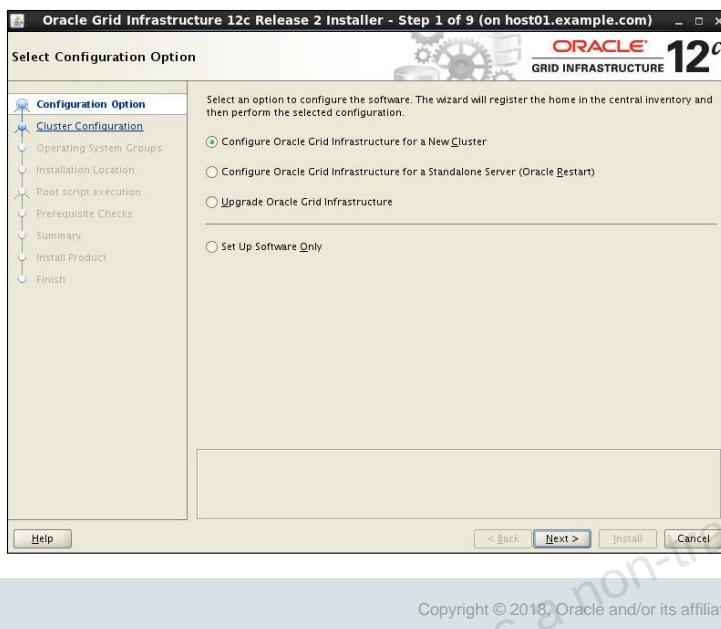


ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Installing Grid Infrastructure

```
$ unzip -d /u01/app/12.2.0/grid /stage/grid_home.zip  
$ gridSetup.sh
```



Starting with Oracle Grid Infrastructure 12c Release 2 (12.2), installation and configuration of Oracle Grid Infrastructure software is simplified with image-based installation.

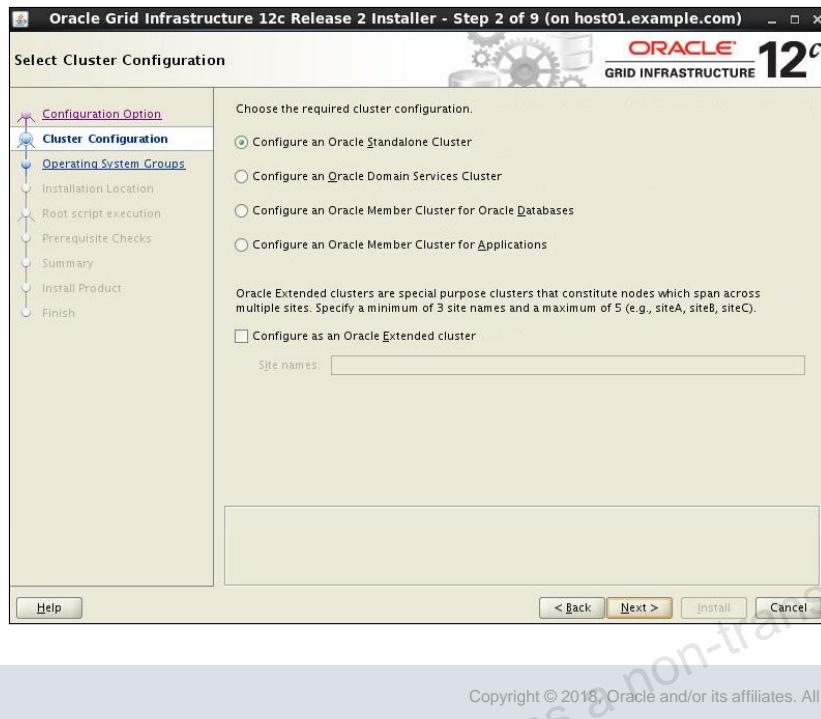
To install Oracle Grid Infrastructure, create the new Grid home with the necessary user group permissions, and then extract the Grid Infrastructure files to the `GRID_HOME` location on the installing node from the zip distribution.

Run the installer by executing the `setupGrid.sh` script as the grid owner from the `GRID_HOME` directory. On the Configuration Option page, select the “Install and Configure Grid Infrastructure for a New Cluster” option and click Next.

Using image-based installation, you can do the following:

- Install and upgrade Oracle Grid Infrastructure for cluster configurations.
- Install Oracle Grid Infrastructure for a standalone server (Oracle Restart).
- Install only Oracle Grid Infrastructure software, and register the software with Oracle inventory.
- Add nodes to your existing cluster, if the Oracle Grid Infrastructure software is already installed or configured.

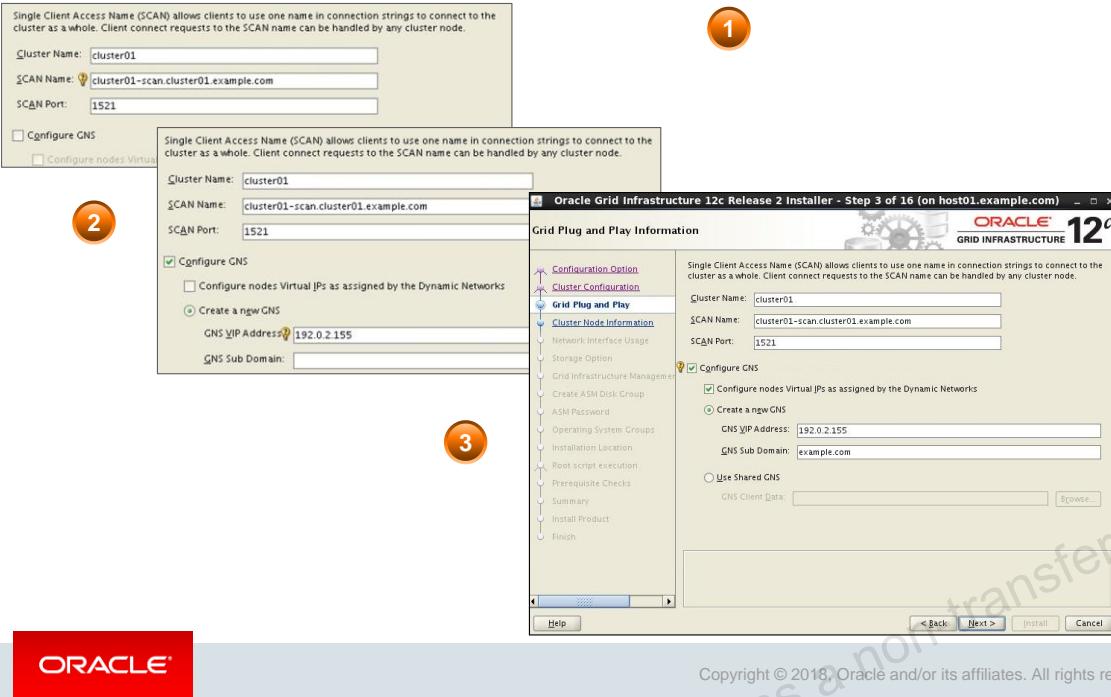
Choosing a Cluster Configuration



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

On the Cluster Configuration page, you can choose to install and configure an Oracle Standalone Cluster, an Oracle Domain Services Cluster, an Oracle Member Cluster for Oracle Databases, or an Oracle Member Cluster for Applications. Select the cluster type required for the installation and click Next. (In this example, an Oracle Standalone Cluster is selected.)

Grid Plug and Play Support



Next, the Grid Plug and Play Information page is displayed. You must supply a cluster name. This name should be unique within your subnet. If Enterprise Manager Cloud Control is used, the cluster name should be unique within the Enterprise Manager management realm.

The single-client access name (SCAN) is the address used by clients that connect to the cluster. SCAN is a domain name registered to three IP addresses, either in the domain name server (DNS) or the Grid Naming Service (GNS). SCAN addresses need to be on the same subnet as the VIP addresses for the nodes in the cluster. The SCAN domain name must be unique within your corporate network. The SCAN port should default to 1521 unless you have a conflict at that port.

Oracle Flex Cluster in 12c Release 1 required the configuration of the GNS because GNS is required to operate Leaf Nodes as part of the cluster. Starting with 12c Release 2, GNS is only required when Leaf Nodes are added to the cluster either during installation or later.

In the first example in the slide, the Configure GNS option is not selected, which means that only Hub Nodes can be added during installation.

In the second example, the Configure GNS option is selected, but the GNS sub Domain field is empty. It is to configure GNS without subdomain delegation. This GNS configuration using static VIP addresses and SCANS enables Oracle Flex Cluster and CloudFS features that require name resolution information within the cluster. However, any node additions or changes must be carried out as manual administration tasks.

The third example in the slide is to employ GNS with subdomain delegation. You select the Configure GNS check box and choose to have DHCP assign VIP addresses by selecting the “Configure nodes Virtual IPs as assigned by the Dynamic Networks” check box. Select “Create a new GNS.” Provide the GNS VIP address and GNS subdomain. When you specify the Cluster Name and the GNS subdomain, SCAN defaults to *clusternname-scan.GNSdomain*. For example, if you start the Grid Infrastructure installation from host03 and the cluster name is `cluster01` and the GNS domain is `example.com`, SCAN becomes `cluster01-scan.cluster01.example.com`.

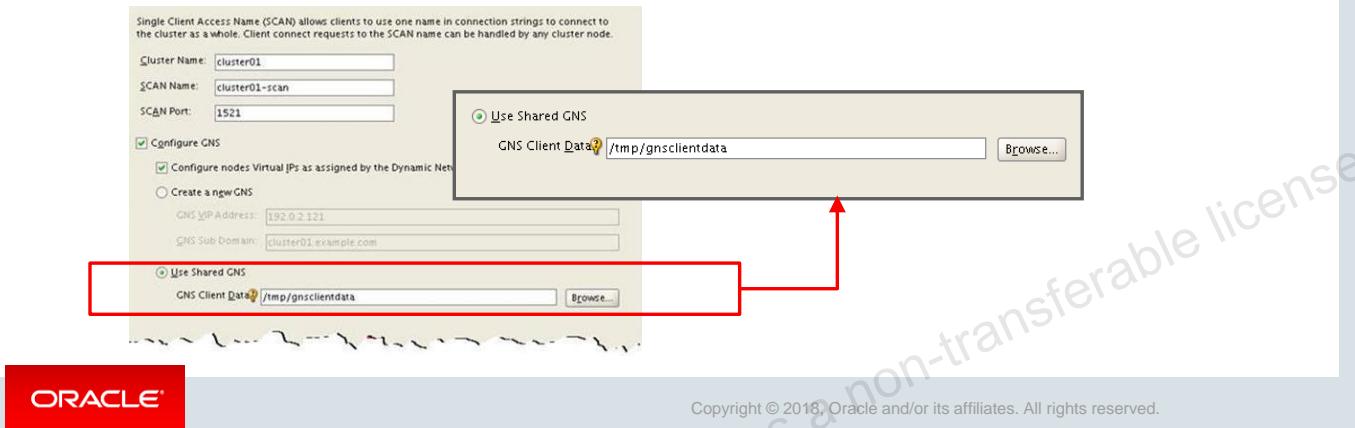
To use an existing GNS, select “Use Shared GNS”

Configuring Shared GNS

1. Export GNS credentials on the GNS server cluster.

```
# srvctl export gns -clientdata /tmp/gnscientdata
```

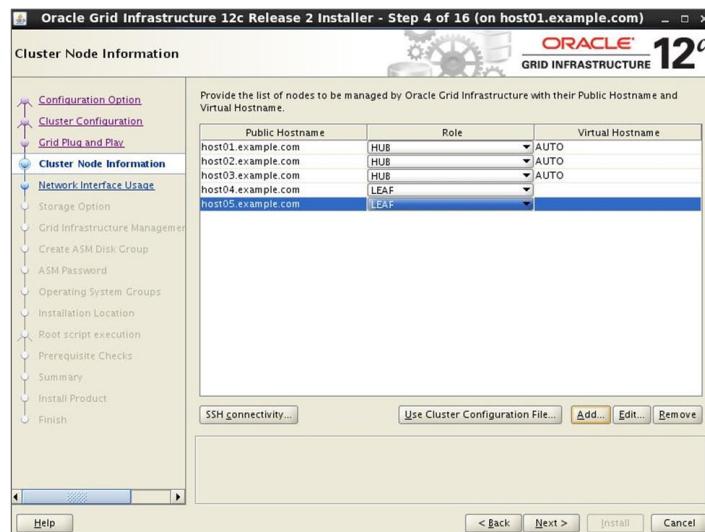
2. Copy the credentials file to the client cluster.
3. Select “Use Shared GNS” and reference the credentials file in Oracle Universal Installer.



If you choose to configure Shared GNS, you will be required to provide the GNS client data that is exported from another host that is serviced by the existing GNS. The configuration process for a client cluster involves registering the client cluster with the GNS server. To perform the registration, a set of GNS server credentials is required. The overall procedure for configuring a GNS client cluster is as follows:

1. Export the GNS credentials on the server cluster by using the `srvctl export gns` command and by specifying a file to store the GNS credentials.
2. Copy the credentials file to the client cluster by using a network file copy (`ftp` or `scp`, for example) or a physical file transfer (CD, DVD, or memory stick, for example).
3. On the Oracle Universal Installer “Grid Plug and Play Information” screen, select “Use Shared GNS” and specify the credentials file in the GNS Client Data field. Click Next to continue.

Cluster Node Information

**ORACLE®**

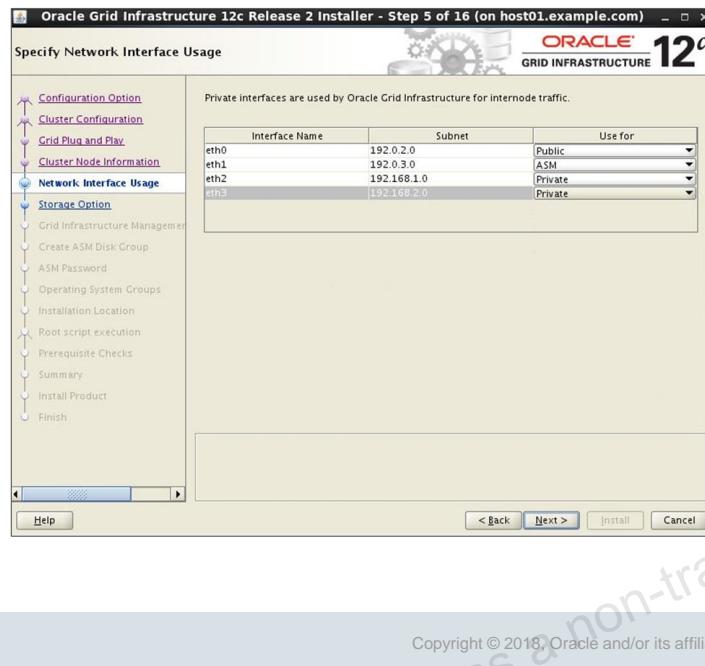
Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

On the Cluster Node Information page, the public host names and virtual host names are defined. In the past, the host and VIP names and addresses were defined in the DNS or locally in a hosts file. If you want to configure your cluster in this manner, make sure that the Configure GNS check box is deselected on the previous screen. Click the Add button and enter the Public Node Name and Virtual Host Name for each node in the cluster.

If GNS is selected on the previous screen and DHCP is configured in the subdomain in which the cluster resides, configuration is simplified. Click the Add button, and then add the host name as shown in the graphic in the slide. There is no need to provide a Virtual IP name for each node because Virtual IP names are automatically configured by Clusterware by using the IPs assigned by DHCP. If you are creating a Flex Cluster, you will need to indicate whether the node that you are adding will be a Hub or Leaf node. Use the drop-down list in the Node Role field to specify this.

Secure Shell (SSH) can be configured for the specified nodes by clicking the SSH Connectivity button. You will be required to provide the software owners password that is common to all nodes. When SSH connectivity has been established, click Next to continue.

Specify Network Interface Usage

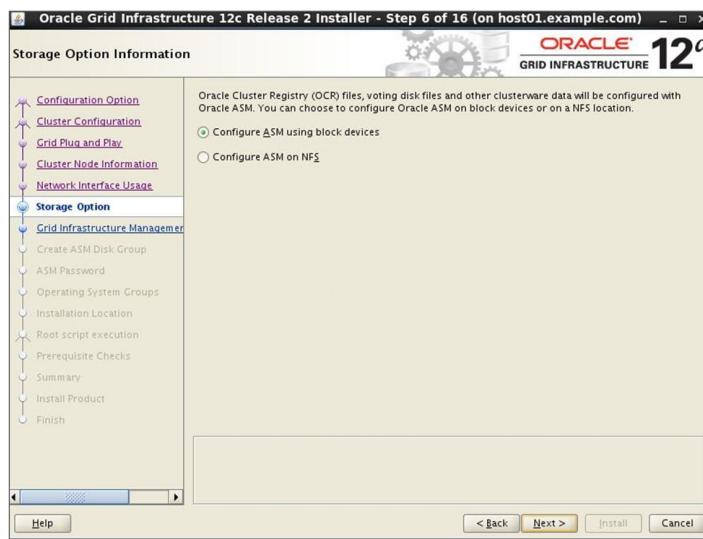


Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

On the Specify Network Interface Usage page, you can select the network interfaces on your cluster nodes to use for internode communication. You must choose one interface for the public network and one for the ASM and the private network.

Ensure that the network interfaces that you choose for the interconnect have enough bandwidth to support the cluster and RAC-related network traffic. To configure the interfaces, click the drop-down list to the right of the interface under the “Use for” column, and select proper usage for each network interface. In the example in the slide, there are four interfaces: eth0, eth1, eth2, and eth3. The eth0 interface is the hosts’ primary network interface and should be marked Public. The eth1 interface is configured for ASM and eth2 and eth3 interfaces are configured for private network. If you have other adapters dedicated to a storage network, they should be marked Do Not Use. When you finish, click the Next button to continue.

Storage Option Information

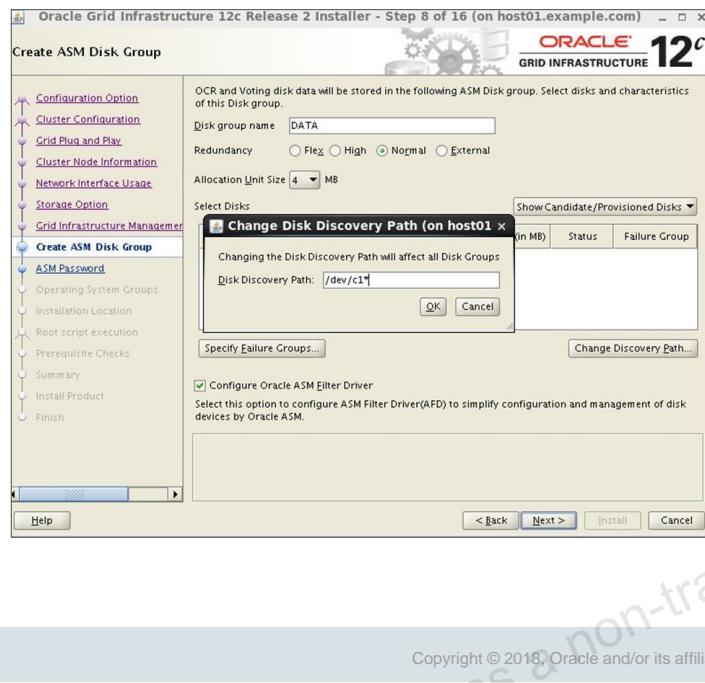


ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

On the Storage Option Information page, you can choose to configure ASM by using block devices or NFS for the initial clusterware installation.

Create ASM Disk Group



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

On the Create ASM Disk Group page, you can configure the default DATA disk group that is used to store the OCR and voting disks. Here, you can change the disk group name, set redundancy, configure failure groups, and set the allocation unit size. If the default discovery path matches no available shared volumes, you can change it here. After the discovery path is properly set, you can choose the required disks for the disk group.

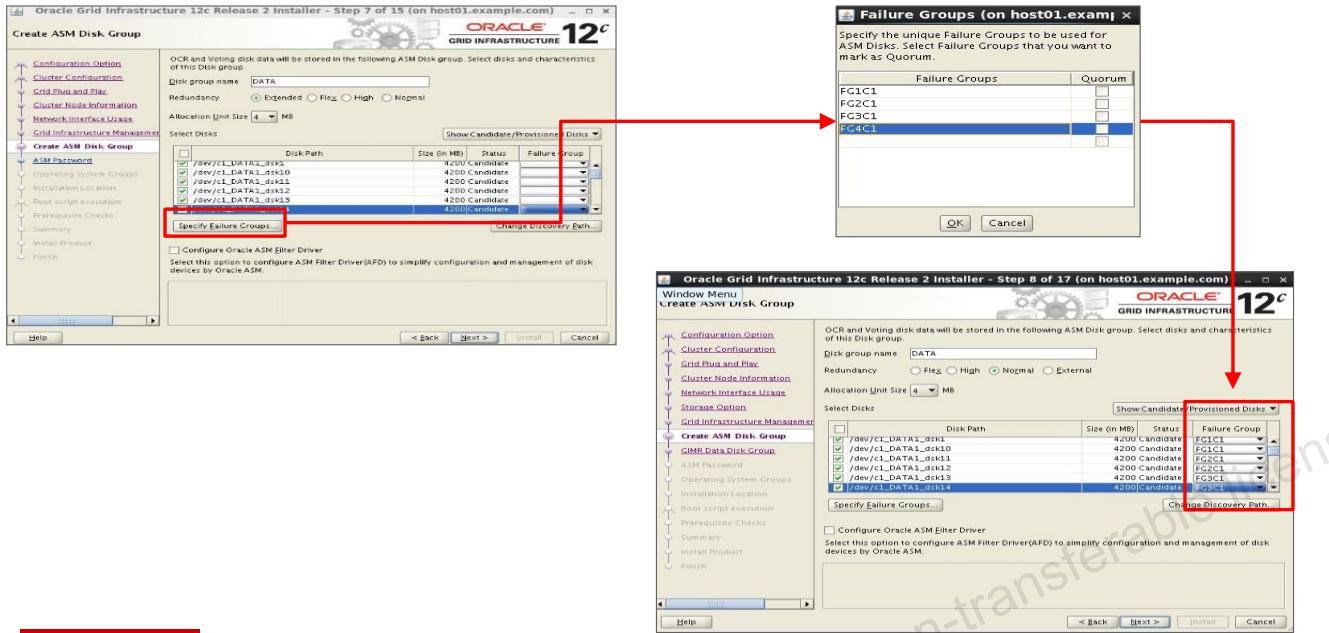
When using ASM, the redundancy for the OCR and voting files is tied to the redundancy that you define for the disk group and the number of disks that you put in the disk group. For Clusterware files, a normal redundancy disk group requires a minimum of three disk devices (two of the three disks are used by failure groups and all three disks are used by the quorum failure group) and provides three voting files and one OCR and mirror of the OCR. When using a normal redundancy disk group, the cluster can survive the loss of one failure group. For most installations, Oracle recommends that you select normal redundancy.

In a high redundancy disk group, Oracle ASM uses three-way mirroring to increase performance and provide the highest level of reliability. High redundancy disk groups provide five voting disk files, one OCR, and three copies (one primary and two secondary mirrors). With high redundancy, the cluster can survive the loss of two failure groups. Although high redundancy disk groups do provide a high level of data protection, you should consider the greater cost of additional storage devices before deciding to select high redundancy disk groups.

A high redundancy disk group requires a minimum of three disk devices (or three failure groups). The effective disk space in a high redundancy disk group is one-third the sum of the disk space in all of its devices.

When you have made all the necessary selections, click Next to continue.

Create ASM Disk Group: Specify Failure Groups

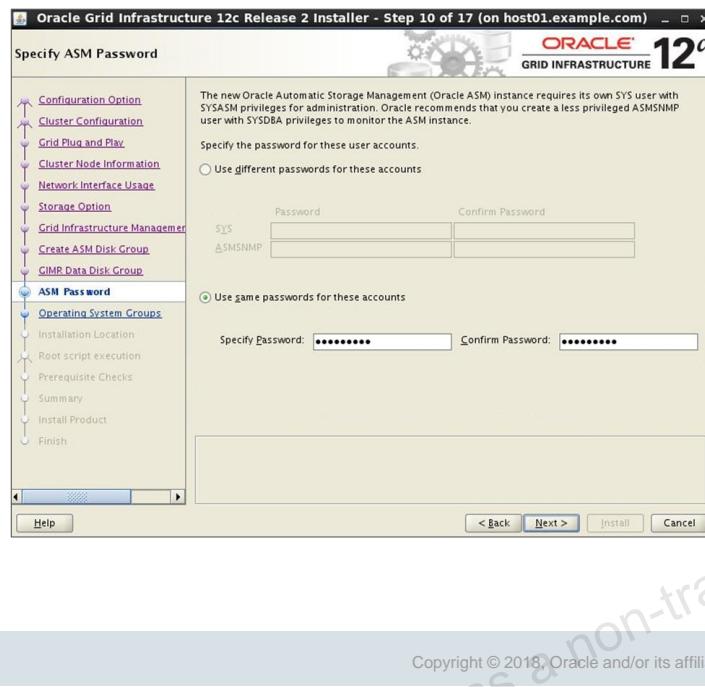


Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

You can configure failure groups on the Create ASM Disk Group page. Click the Specify Failure Group button, and then add individual failure groups in the Failure Group input window. You can also choose to mark failure groups as quorum, if you wish. A quorum failure group, which is a special type of failure group, contains mirror copies of voting files when voting files are stored in normal or high redundancy disk groups.

The quorum failure group is used to ensure that a quorum of the specified failure groups is available. When Oracle ASM mounts a disk group that contains Oracle Clusterware files, the quorum failure group is used to determine if the disk group can be mounted in the event of the loss of one or more failure groups. Disks in the quorum failure group do not contain user data; therefore, a quorum failure group is not considered when determining redundancy requirements with respect to storing user data.

Specify ASM Password

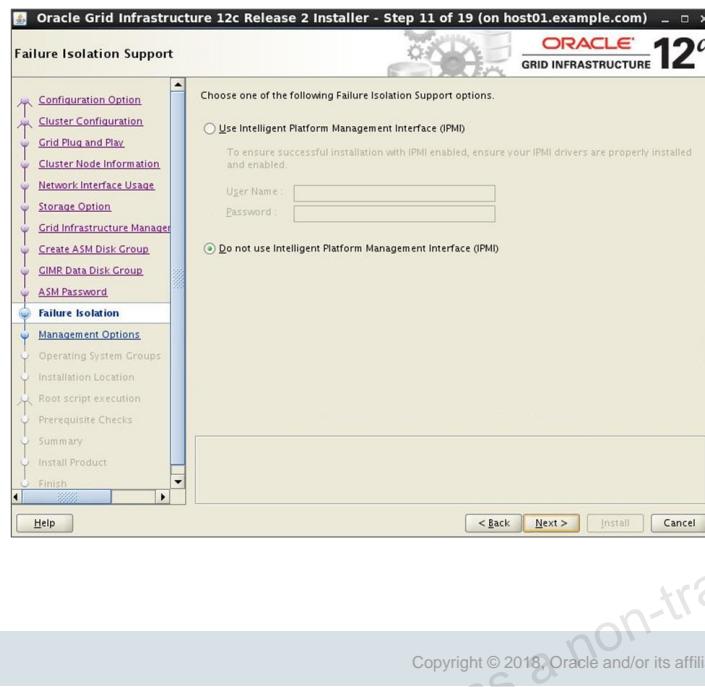


ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

On the Specify ASM Password page, you specify the **SYS** and **ASMSNMP** passwords for the ASM instances. You can use different passwords for the two accounts, or you can use the same password if you like. When you have finished, click **Next** to continue.

Failure Isolation Support with IPMI



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

ORACLE®

The Intelligent Platform Management Interface (IPMI) provides a set of common interfaces to computer hardware and firmware that administrators can use to monitor system health and manage the system. Oracle Clusterware can integrate IPMI to provide failure isolation support and to ensure cluster integrity.

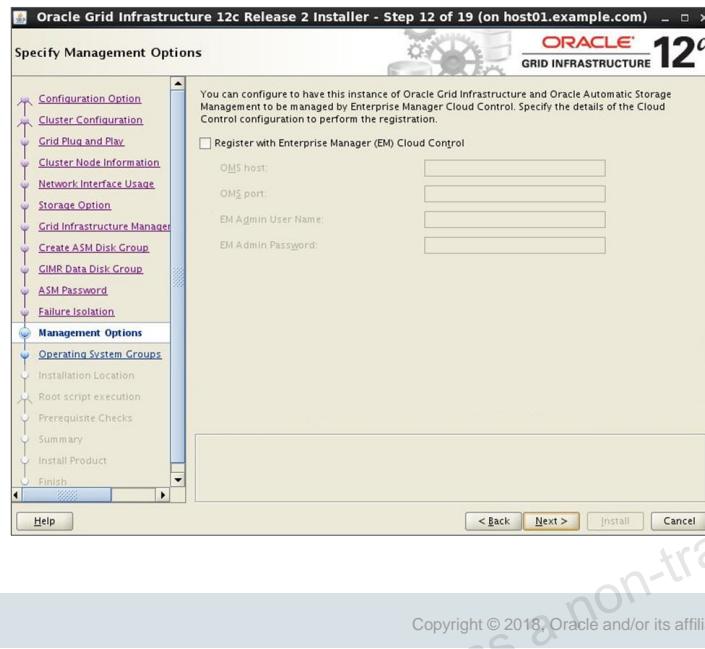
You must have the following hardware and software configured to enable cluster nodes to be managed with IPMI:

- Each cluster member node requires a Baseboard Management Controller (BMC) that runs firmware that is compatible with IPMI version 1.5 or later, which supports IPMI over local area networks (LANs) and is configured for remote control.
- Each cluster member node requires an IPMI driver installed on each node.
- The cluster requires a management network for IPMI. This can be a shared network, but Oracle recommends that you configure a dedicated network.
- Each cluster member must be connected to the management network.
- Some server platforms put their network interfaces into a power saving mode when they are powered off. In this case, they may operate only at a lower link speed (for example, 100 MB, instead of 1 GB). For these platforms, the network switch port to which the BMC is connected must be able to auto-negotiate down to the lower speed, or IPMI cannot function properly.

- If you intend to use IPMI, you must provide an administration account username and password when prompted during installation.

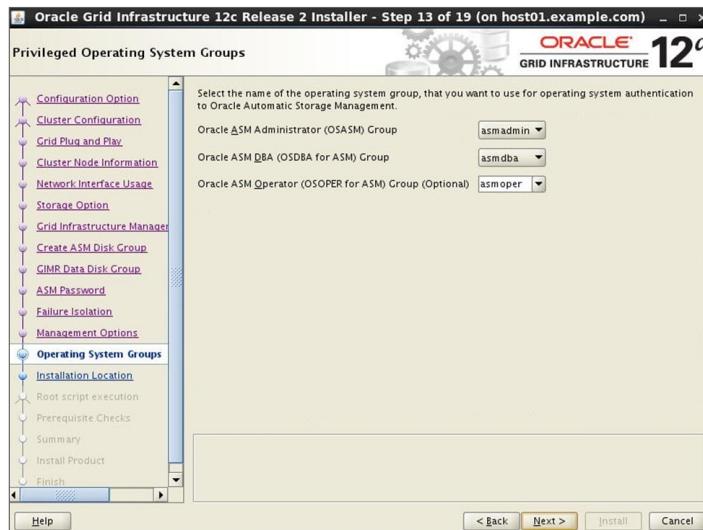
Note: For Oracle Clusterware to communicate with BMC, the IPMI driver must be installed permanently on each node, so that it is available on system restarts.

Specify Management Options



On the Specify Management Options page, you can configure the cluster's Clusterware and ASM instances to communicate with Enterprise Manager Cloud Control. You will need to provide the OMS host and port number and the Enterprise Manager admin username and password.

Privileged Operating System Groups

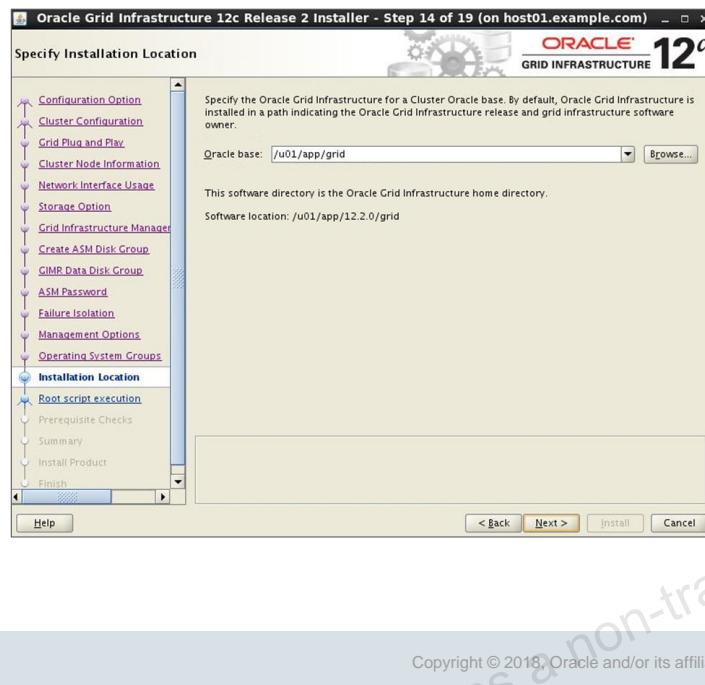


ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

On the Privileged Operating System Groups page, you must specify the groups that the Grid Infrastructure owner should belong to for proper operating system authentication to ASM. The example in the slide specifies `asmdba` for the ASM Database Administrator (OSDBA) group, `asmoper` for the ASM Instance Administration Operator (OSOPER) group, and `asmadmin` for the ASM Instance Administrator (OSASM) group. Click Next to continue.

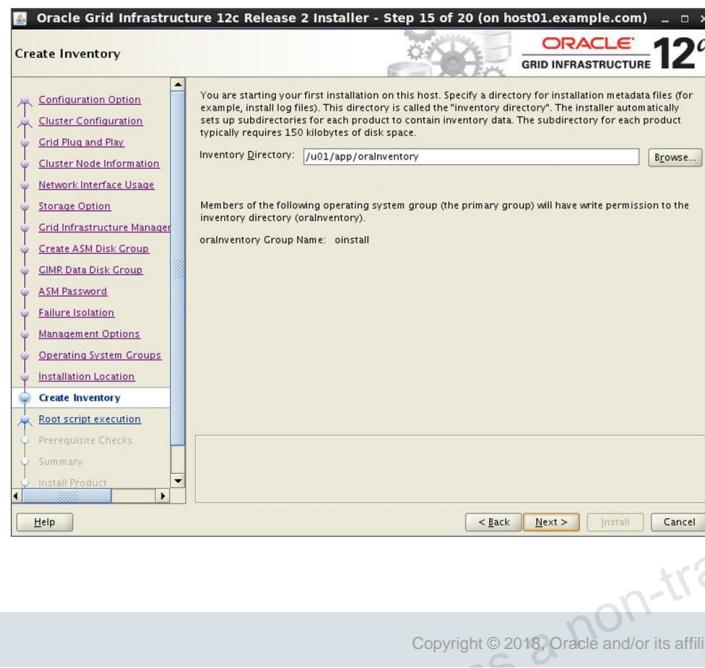
Specify Installation Location



On the Specify Installation Location page, enter a base location for installing Oracle software. A separate Oracle base should be created for each OS user that owns an Oracle software installation. In the example in the slide, the location is /u01/app/grid.

In the Software Location field, enter a directory in which to install Oracle Grid Infrastructure. In the example, the grid operating system user will own this installation, so the software location is /u01/app/12.2.0/grid. When you have entered proper Oracle Base and Software Location values, click Next to continue.

Create Inventory

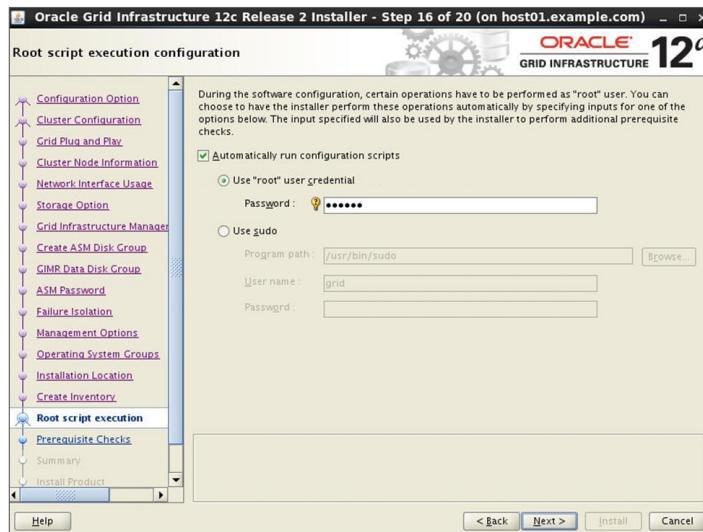


ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

On the Create Inventory page, enter a location under your Oracle Base to store the Oracle Inventory or accept the default value. Click Next to continue.

Root Script Execution Configuration

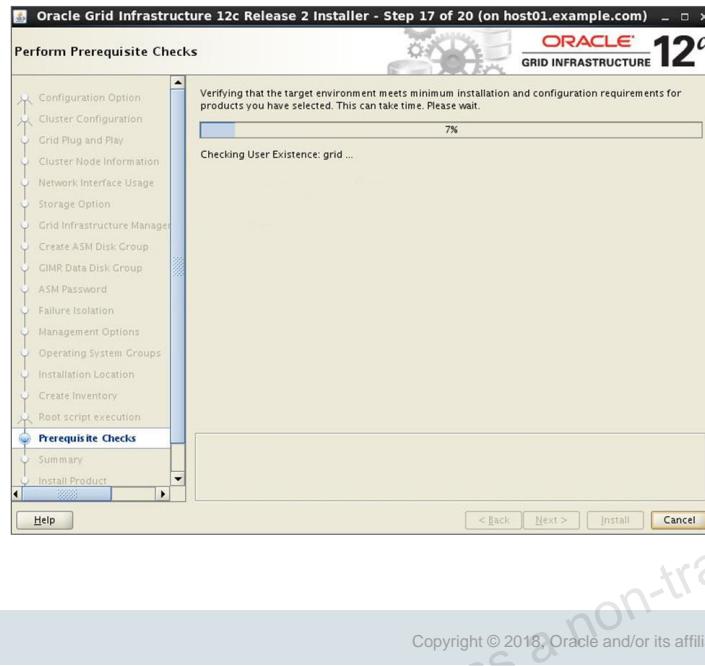


ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

On the “Root script execution configuration” screen, you can choose how to run the root scripts by selecting the “Automatically run configuration scripts” check box. If you choose to run them as the `root` user, you must provide the root password. Alternatively, you can choose to use `sudo` to run the root scripts. You must provide a privileged username and password and that user must have permission to run both the `root.sh` and `orainstRoot.sh` scripts as defined in the `/etc/sudoers` file. If you do not select the “Automatically run configuration scripts” check box, you will be prompted to run the `root` scripts near the end of the installation as in previous releases.

Perform Prerequisite Checks



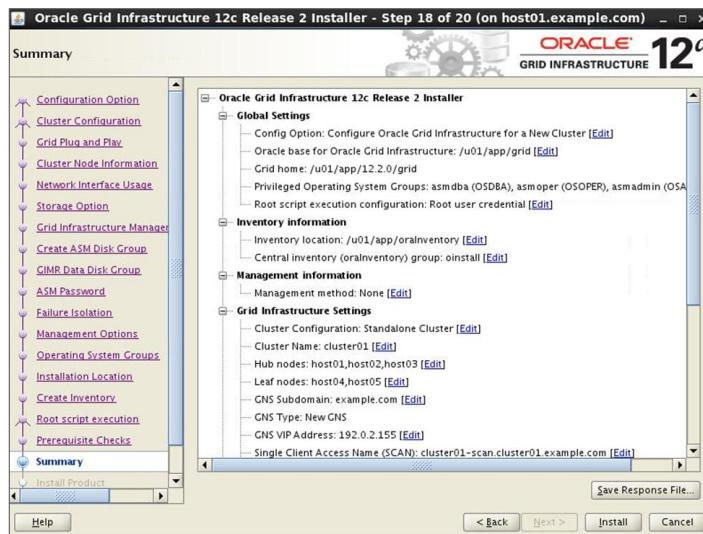
Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

ORACLE®

On the Perform Prerequisite Checks page, the OUI provides additional guidance to ensure recommended deployment and to prevent configuration issues. In addition, configuration assistants validate configurations and provide scripts to fix issues that you can use or choose to reject. Clicking the Fix & Check Again button opens a window instructing you to run a fixup script that must be run as `root` on all nodes before continuing.

The installer can determine whether the deficiency can be corrected, and presents the user performing the installation with the option of allowing the OUI to make the required correction. When you click the Fix & Check Again button, a script is generated on the nodes where the deficient condition exists. After executing the scripts as `root` on the nodes, the kernel parameter is adjusted, allowing the installation to continue uninterrupted. When the prerequisite checks have completed successfully, click Next to continue. It is possible that an installation deficiency cannot be corrected with a generated fixup script. The installer will indicate this with a “No” in the Fixable column for that item.

Install Product



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

When the Summary page appears, check the installation information displayed. If the information is correct, click Finish to begin the software setup. You can monitor the progress of the Grid Infrastructure installation on the Install Product screen.

Verifying the Grid Infrastructure Installation

```
[grid@host01 ~]$ crsctl stat res -t
Name          Target  State       Server           State details
Local Resources
ora.ASMNET1LSNR ASM.lsnr
    ONLINE  ONLINE   host01           STABLE
    ONLINE  ONLINE   host02           STABLE
ora.DATA.dg
    ONLINE  ONLINE   host01           STABLE
    ONLINE  ONLINE   host02           STABLE
...
Cluster Resources
ora.LISTENER_SCAN1.lsnr
    1      ONLINE  ONLINE   host01           STABLE
ora.LISTENER_SCAN2.lsnr
    1      ONLINE  ONLINE   host02           STABLE
ora.LISTENER_SCAN3.lsnr
    1      ONLINE  ONLINE   host02           STABLE
ora.MGMTLSNR
    1      ONLINE  ONLINE   host02           169.254.18.156 192.1
                                         68.1.102,STABLE
ora.asm
    1      ONLINE  ONLINE   host01           STABLE
    2      ONLINE  ONLINE   host02           STABLE
    3      OFFLINE OFFLINE
ora.cvu
    1      ONLINE  ONLINE   host02           STABLE
...
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Execute the `crsctl` command as shown in the slide to confirm that all cluster and local resources are up and running. If you elected to incorporate GNS in your Grid Infrastructure installation, you should confirm that your DNS is properly forwarding address requests for your application and SCAN VIPs to your GNS and that they are resolved properly. You can do this with `host`. Execute the `host` command as shown:

```
# host -a host01-vip.cluster01.example.com
;; QUESTION SECTION:
;host01-vip.cluster01.example.com. IN A
;; ANSWER SECTION:
host01-vip.cluster01.example.com. 120 IN A 192.0.2.237
...
# host -a cluster01-scan.cluster01.example.com
;; QUESTION SECTION:
;cluster01-scan.cluster01.example.com. IN A
;; ANSWER SECTION:
cluster01-scan.cluster01.example.com. 120 IN A 192.0.2.232
cluster01-scan.cluster01.example.com. 120 IN A 192.0.2.239
cluster01-scan.cluster01.example.com. 120 IN A 192.0.2.240
;; AUTHORITY SECTION:
cluster01.example.com. 86400 IN NS cluster01-
gns.example.com.
...
```

Understanding Offline Processes

- Grid Infrastructure provides required resources for various Oracle products and components.
- Some of those products and components are optional, so you can install and enable them later.
- Grid Infrastructure preconfigures and registers resources for these, activating them only when you add them.
- As a result, some components may be listed as OFFLINE after the installation.

```
[grid@host01 ~]$ crsctl stat res -t
...
ora.LISTENER_LEAF.lsnr
    OFFLINE OFFLINE      host04          STABLE
```

- Resources listed as TARGET:OFFLINE and STATE:OFFLINE represent components that are registered but not enabled.
 - These need not be monitored and they use no resources.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Grid Infrastructure provides required resources for various Oracle products and components. Some of those products and components are optional, so you can install and enable them after installing Oracle Grid Infrastructure. To simplify postinstall additions, Oracle Grid Infrastructure preconfigures and registers all required resources for all products available for these products and components, but only activates them when you choose to add them. As a result, some components may be listed as OFFLINE after the installation of Oracle Grid Infrastructure.

Resources listed as TARGET:OFFLINE and STATE:OFFLINE do not need to be monitored. They represent components that are registered, but not enabled, so they do not use any system resources. If an Oracle product or component is installed on the system and it requires a particular resource to be online, the software prompts you to activate the required offline resource.

Check ASM Function for Oracle Clusterware Files

- After Oracle Grid Infrastructure installation, Oracle Clusterware files are stored on Oracle ASM, check it with the `srvctl status asm` command:

```
[grid@host01 ~]$ srvctl status asm -detail
ASM is running on host02,host03,host01
ASM is enabled.
ASM instance +ASM1 is running on node host01
Number of connected clients: 3
Client names: -MGMTDB:_mgmtdb:cluster01 host01.example.com:_OCR:cluster01
orcl_3:orcl:cluster01
ASM instance +ASM2 is running on node host03
Number of connected clients: 2
Client names: host03.example.com:_OCR:cluster01 orcl_2:orcl:cluster01
ASM instance +ASM3 is running on node host02
Number of connected clients: 2
Client names: host02.example.com:_OCR:cluster01 orcl_1:orcl:cluster01
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

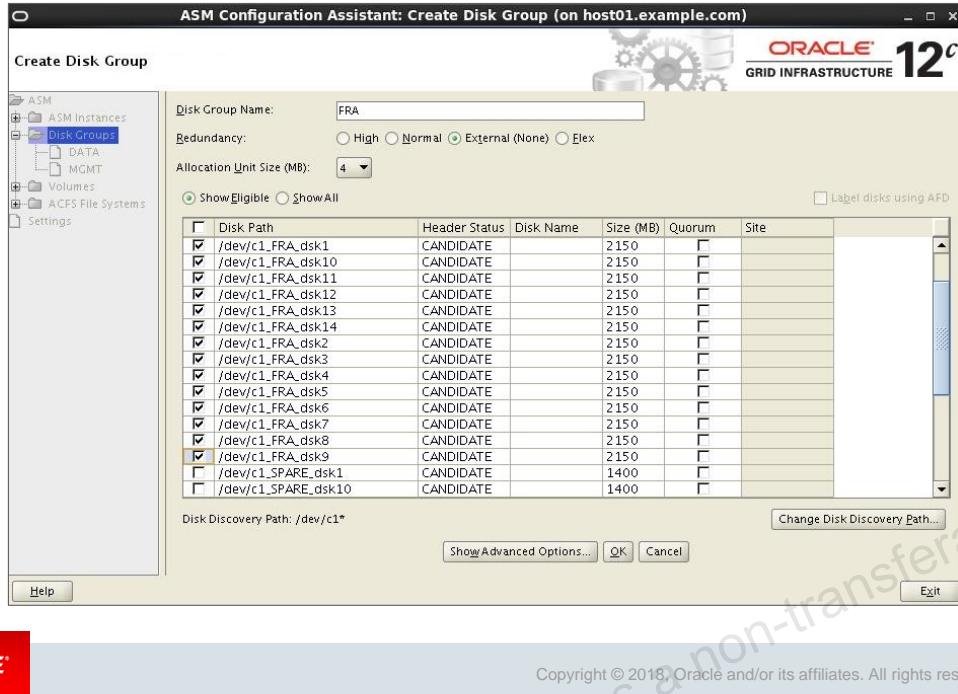
Confirm Oracle ASM is running after installing Oracle Grid Infrastructure.

After Oracle Grid Infrastructure installation, Oracle Clusterware files are stored on Oracle ASM. Use the following command syntax as the Oracle Grid Infrastructure installation owner (grid) to confirm that your Oracle ASM installation is running:

```
$ srvctl status asm [-detail]
```

Note: To manage Oracle ASM or Oracle Net 11g Release 2 (11.2) or later installations, use the `srvctl` binary in the Oracle Grid Infrastructure home for a cluster (Grid home). If you have Oracle Real Application Clusters or Oracle Database installed, then you cannot use the `srvctl` binary in the database home to manage Oracle ASM or Oracle Net.

Create a Fast Recovery Area Disk Group



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

During installation, by default you can create one disk group. If you plan to add an Oracle Database for a stand-alone server or an Oracle RAC database, you should create a disk group for the Fast Recovery Area for database files.

The Fast Recovery Area is a unified storage location for all Oracle Database files related to recovery. Database administrators can define the `DB_RECOVERY_FILE_DEST` parameter to the path for the Fast Recovery Area to enable on-disk backups, and rapid recovery of data. Enabling rapid backups for recent data can reduce requests to system administrators to retrieve backup tapes for recovery operations. When you enable Fast Recovery in the `init.ora` file, all RMAN backups, archive logs, control file automatic backups, and database copies are written to the Fast Recovery Area. RMAN automatically manages files in the Fast Recovery Area by deleting obsolete backups and archive files no longer required for recovery.

To create a Fast Recovery Area Disk Group, perform the following steps:

1. As the Grid installation owner, navigate to the Grid home bin directory, and start ASMCA:
 \$ cd /u01/app/12.2.0/grid/bin
 \$./asmca
2. ASMCA opens at the Disk Groups tab. Click Create to create a new disk group. The Create Disk Groups window opens. In the Disk Group Name field, enter a descriptive name for the Fast Recovery Area group, for example, FRA. In the Redundancy section, select the level of redundancy you want to use. In the Select Member Disks field, select eligible disks to be added to the Fast Recovery Area, and click OK.
3. The Disk Group Creation window opens to inform you when disk group creation is complete. Click OK.
4. Click Exit.

Modifying Oracle Clusterware Binaries After Installation

After installation, if you need to modify the Oracle Clusterware configuration, you must unlock the Grid home.

1. Log in as root, go to `<Grid_home>/crs/install`, and unlock the Grid home by using the following command:

```
# perl rootcrs.pl -unlock -crshome /u01/app/12.2.0/grid
```

2. As the Grid software owner, relink binaries. This example updates the interconnect protocol from UDP to IPC:

```
# su - grid  
$ make -f $ORACLE_HOME/rdbms/lib/ins_rdbms.mk ipc_rds ioracle
```

3. Relock the Grid home and restart the cluster by using the following command: `# perl rootcrs.pl -patch`.

4. Repeat steps 1 through 3 on each cluster member node.

ORACLE

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

After installation, if you need to modify the Oracle Clusterware configuration, then you must unlock the Grid home. For example, if you want to apply a one-off patch, or if you want to modify an Oracle Exadata configuration to run IPC traffic over RDS on the interconnect instead of using the default UDP, you must unlock the Grid home. Unlock the home using the following procedure:

1. Log in as root, and change directory to the `Grid_home/crs/install` path, where `Grid_home` is the path to the Grid home, and unlock the Grid home by using the `rootcrs.pl -unlock -crshome Grid_home` command, where `Grid_home` is the path to your Grid infrastructure home. For example, with the Grid home `/u01/app/12.2.0/grid`, enter the following command:
`# cd /u01/app/12.2.0/grid/crs/install`
`# perl rootcrs.pl -unlock -crshome /u01/app/12.2.0/grid`
2. Change user to the Oracle Grid Infrastructure software owner and relink binaries using the command syntax `make -f Grid_home/rdbms/lib/ins_rdbms.mk target`, where `Grid_home` is the Grid home and `target` is the binaries that you want to relink. For example, where the `grid` user is `grid`, `$ORACLE_HOME` is set to the Grid home, and where you are updating the interconnect protocol from UDP to IPC, you enter the following command:
`# su grid`
`$ make -f $ORACLE_HOME/rdbms/lib/ins_rdbms.mk ipc_rds ioracle`

3. Relock the Grid home and restart the cluster by using the following command:

```
# perl rootcrs.pl -patch
```
4. Repeat steps 1 through 3 on each cluster member node.

Unconfiguring Oracle Clusterware Without Removing Binaries

To unconfigure Oracle Clusterware:

1. Log in as the root user on a node where you encountered an error.

2. Change directory to `<Grid_home>/crs/install`.

Example: # cd /u01/app/12.2.0/grid/crs/install

3. Run `rootcrs.pl` with the `-deconfig` and `-force` flags. Example: # perl rootcrs.pl -deconfig -force
(Repeat on other nodes as required.)

4. If you are deconfiguring Oracle Clusterware on all nodes in the cluster, enter the following command on the last node:

perl rootcrs.pl -deconfig -force -lastnode

The `-lastnode` flag completes deconfiguration of the cluster, including the OCR and voting disks.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Running the `rootcrs.pl -deconfig -force` command enables you to unconfigure Oracle Clusterware on one or more nodes without removing installed binaries. This feature is useful if you encounter an error on one or more cluster nodes during installation when running the `root.sh` command, such as a missing operating system package on one node. By running `rootcrs.pl -deconfig -force` on nodes where you encounter an installation error, you can unconfigure Oracle Clusterware on those nodes, correct the cause of the error, and then run `root.sh` again.

To unconfigure Oracle Clusterware:

1. Log in as the root user on a node where you encountered an error.

2. Change directory to `Grid_home/crs/install`.

Example: # cd /u01/app/12.2.0/grid/crs/install

3. Run `rootcrs.pl` with the `-deconfig` and `-force` flags.

Example: # perl rootcrs.pl -deconfig -force

(Repeat on other nodes as required.)

4. If you are deconfiguring Oracle Clusterware on all nodes in the cluster, enter the following command on the last node:

perl rootcrs.pl -deconfig -force -lastnode

The `-lastnode` flag completes deconfiguration of the cluster, including the OCR and voting disks.



Quiz

You must configure GNS to take advantage of automatic VIP configuration.

- a. True
- b. False



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Perform an image-based Grid Infrastructure Installation
- Verify the installation
- Configure Automatic Storage Management (ASM) disk groups



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Practice 5: Overview

This practice covers installing Oracle Grid Infrastructure to create a Standalone (Flex) Cluster.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Managing Cluster Nodes

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Perform the prerequisite steps to extend a cluster
- Use `addnode.sh` to add a node to a cluster
- Delete a node from a cluster

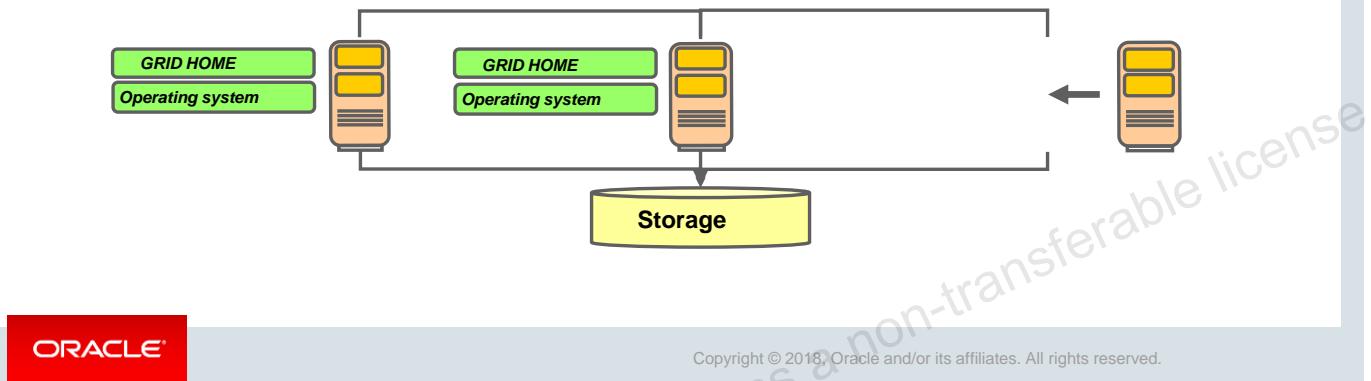


Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Adding a Cluster Node

There are three methods you can use to add a node to your cluster:

- Using Rapid Home Provisioning
- Using Oracle Grid Infrastructure Installer
- Using `addnode.sh` shell script



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

There are three methods you can use to add a node to your cluster:

Using Rapid Home Provisioning to Add a Node: Rapid Home Provisioning and Maintenance (RHP) has evolved significantly since its initial release in Grid Infrastructure 12c Release 1, which focused on provisioning and patching Oracle Database homes. With RHP 12c Release 2, a full range of provisioning and maintenance features are delivered, which includes the Database and Grid Infrastructure provisioning, patching, scaling, and upgrading with a single command.

If you have a Rapid Home Provisioning Server, then you can use Rapid Home Provisioning to add a node to a cluster with one command, as shown in the following example:

```
$ rhpctl addnode gihome -client rhpclient -newnodes clientnode2:clientnode2-vip -root
```

The example adds a node named `clientnode2` with VIP `clientnode2-vip` to the Rapid Home Provisioning Client named `rhpclient`, using root credentials (login for the node you are adding).

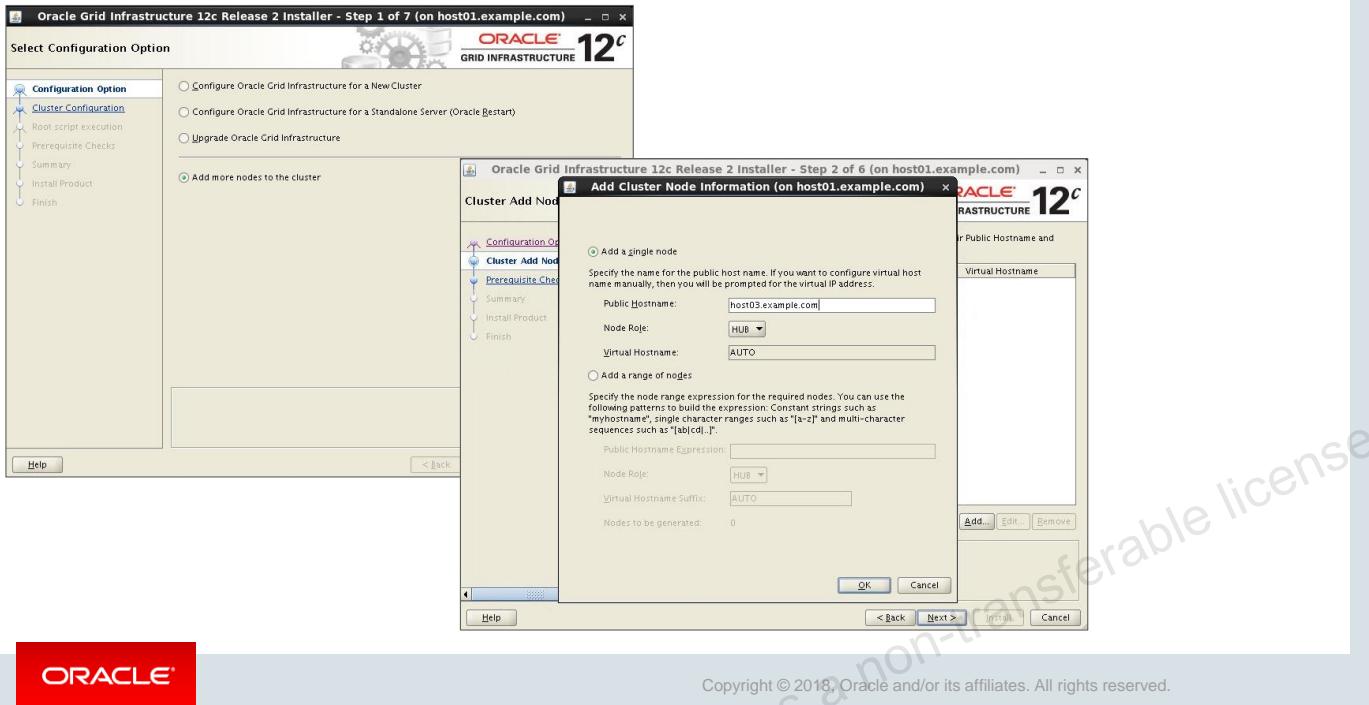
Note: For more information about Rapid Home Provisioning, refer to *Oracle Clusterware Administration and Deployment Guide, 12c Release 2 (12.2)*.

Using Oracle Grid Infrastructure Installer to Add a Node: If you do not want to use Rapid Home Provisioning to add a node to the cluster, then you can use the Oracle Grid Infrastructure installer 12c Release 2 to accomplish the task. To add a node to the cluster using the Oracle Grid Infrastructure installer.

Using addnode.sh to Add Nodes: It Invokes a subset of OUI functionality and is used to add nodes to an existing Oracle Clusterware environment. It runs with and without a graphical interface, but does not perform the prerequisite operating system tasks .

In this lesson, you examine the use of the `addNode.sh` shell script to add a new cluster node to the existing cluster. 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.

Using Oracle Grid Infrastructure Installer: Example



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Using Oracle Grid Infrastructure Installer to Add a Node: The example in the slide shows how to add a node to the cluster using Oracle Grid Infrastructure installer 12c Release 2.

- Run `./gridsetup.sh` to start the installer.
- On the Select Configuration Option page, select Add more nodes to the cluster.
- On the Cluster Node Information page, click Add... to provide information for nodes you want to add.
- When the verification process finishes on the Perform Prerequisite Checks page, check the summary and then click Install.

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 prerequisite tasks for Oracle Clusterware installation:
 - Check system requirements.
 - Check network requirements.
 - Install the required operating system packages.
 - Set kernel parameters.
 - Create groups and users.
 - Create the required directories.
 - Configure the installation owner's shell limits.
 - Configure Secure Shell (SSH) and enable user equivalency.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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.

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

4. Verify the installation with the Cluster Verify utility (`cluvfy`) from existing nodes.

- Perform a post-hardware and operating system check.

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

- 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
```

5. Ensure that the Management Repository has at least an additional 500 MB for each additional node (above four).

```
[grid@host01]$ oclomon manage -get repsize
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

For step 4, you invoke `cluvfy` from an existing node (`host01`) to perform a posthardware and operating system installation check against the new node (`host03`).

Next, run the CVU command from the `Grid_home/bin` directory on an existing node to obtain a detailed comparison of the properties of the reference node with all of the other nodes that are part of your current cluster environment. Specify a comma-delimited list of nodes after the `-n` option. In the following example, `orainventory_group` is the name of the Oracle Inventory group, and `osdba_group` is the name of the OSDBA group:

```
$ cluvfy comp peer [-refnode <ref_node>] -n <node_list>
[-orainv <orainventory_group>] [-osdba <osdba_group>] [-verbose]
```

If errors are discovered while performing these checks, they must be corrected before continuing with the `addNode.sh` script.

In step 5, ensure that the Grid Infrastructure Management Repository has at least an additional 500 MB of space for each node added above four, as follows:

```
$ oclomon manage -get repsize
```

Add additional space, if required, as follows:

```
$ oclomon manage -repos changereposize <total_in_MB>
```

In the slide examples, `host01` is an existing node and `host03` is the node 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 -fixup
```

3. To add host03 to a standard cluster (Pre-12.2):

```
[grid@host01]$ cd <Grid_Home>/addnode
[grid@host01]$ ./addnode.sh "CLUSTER_NEW_NODES={host03}" \
"CLUSTER_NEW_VIRTUAL_HOSTNAMES={host03-vip}"
```

If adding host03 to a Flex cluster, specify the node role:

```
[grid@host01]$ ./addnode.sh -silent "CLUSTER_NEW_NODES={host03}"
"CLUSTER_NEW_VIRTUAL_HOSTNAMES={host03-vip,}" "CLUSTER_NEW_NODE_ROLES={hub|leaf}"
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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 with 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:

```
$ ./addNode.sh -silent "CLUSTER_NEW_NODES={host03}"
"CLUSTER_NEW_VIRTUAL_HOSTNAMES={host03-vip}"
```

If GNS is implemented in the cluster, add `host03` as follows:

```
$ ./addNode.sh "CLUSTER_NEW_NODES={host03}"
```

If you are adding the new host to an Oracle Flex cluster, make sure you specify the node role when running `addnode.sh`. Remember that Hub Nodes always have VIPs but Leaf Nodes may not. If you want to add multiple nodes by using the syntax in the slide, you can use syntax similar to the following, where `host03` is a Hub Node and `host04` is a Leaf Node:

```
$ ./addnode.sh -silent "CLUSTER_NEW_NODES={host03,host04}"
"CLUSTER_NEW_VIRTUAL_HOSTNAMES={host03-vip,}"
"CLUSTER_NEW_NODE_ROLES={hub,leaf}"
```

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/12.2.0/grid/root.sh #On nodes host03
```

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

Completing OUI Silent Node Addition

4. Perform integrity checks on the cluster.

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



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Adding a Node to a Cluster on Windows Systems

To add a node called `host03` to a cluster on a Windows-based cluster comprising `host01` and `host02`:

1. Verify the integrity of the new cluster node, `host03`:

```
C:\>cluvfy stage -pre nodeadd -n host03 -fixup
```

2. On `host01`, run the `addnode.bat` script from the `GRID_HOME\addnode` directory:

```
C:\>addnode.bat "CLUSTER_NEW_NODES={host03}"  
"CLUSTER_NEW_VIRTUAL_HOSTNAMES={host03-vip}"
```

3. Run the following command on the new node:

```
C:\>Grid_home\crs\config\gridconfig.bat
```

4. Verify the integrity of the Clusterware components on all configured nodes, new and existing:

```
C:\>cluvfy stage -post crsinst -n all -verbose
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Ensure that you complete the prerequisites listed earlier in this lesson before adding nodes. This procedure describes how to add a node to your cluster. This procedure assumes that:

- There is an existing cluster with two nodes named `host01` and `host02`
- You are adding a node named `host03`
- You have successfully installed Oracle Clusterware on `host01` and `host02` in a local home where `Grid_home` represents the successfully installed home

To add a node to your Windows cluster, perform the following steps:

1. Verify the integrity of the new cluster node `host03`:

```
C:\>cluvfy stage -pre nodeadd -n host03 [-fixup] [-verbose]
```

2. On `host01`, go to the `Grid_home\addnode` directory and run the `addnode.bat` script:

```
C:\>addnode.bat "CLUSTER_NEW_NODES={host03}"  
"CLUSTER_NEW_VIRTUAL_HOSTNAMES={host03-vip}"
```

3. Run the following command on the new node:

```
C:\>Grid_home\crs\config\gridconfig.bat
```

4. Run the following command to verify the integrity of the Oracle Clusterware components on all of the configured nodes, both the preexisting nodes and the nodes that you have added:

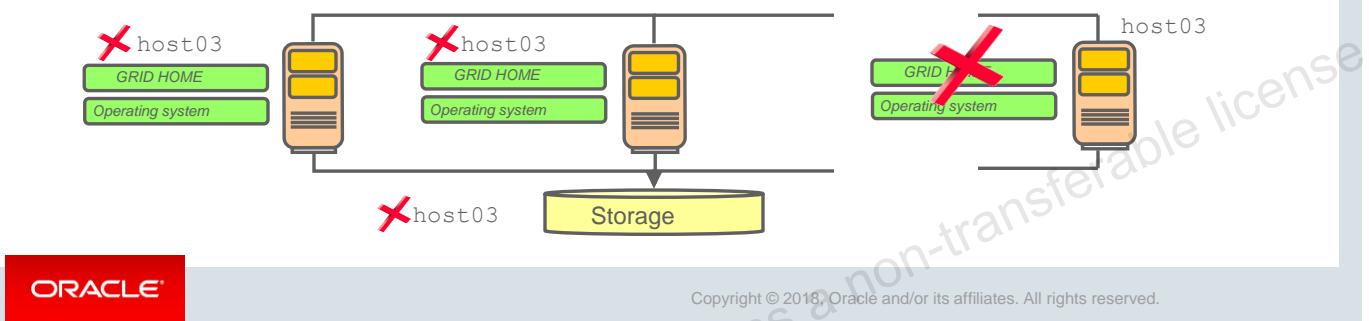
```
C:\>cluvfy stage -post crsinst -n all [-verbose]
```

After you complete the procedure in this section for adding nodes, you can optionally extend Oracle Database with Oracle RAC components to the new nodes, making them members of an existing Oracle RAC database.

Deleting a Node from the Cluster

A series 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®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Deleting a Node from the Cluster

1. Verify the location of the Oracle Clusterware home.
2. From a node that will remain, run the following as `root` to expire the Cluster Synchronization Service (CSS) lease on the node that you are deleting:

```
[root@host01]# crsctl unpin css -n host03
```

3. Run the following command on the node being deleted. **DO NOT** forget the `-local` flag.

```
[grid@host03]$ <Grid_home>/deinstall/deinstall -local
```

4. From any remaining node, run the following command from the `<Grid_home>/bin` directory as `root` to delete the node from the cluster:

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



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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.

```
[root@host01 ~]# olsnodes -s -t
host01 Active Unpinned
host03 Active Unpinned
host02 Active Unpinned
host04 Active Unpinned
host05 Active Unpinned
```

Step 3: For a local home, deinstall the Oracle Clusterware home from the node that you want to delete. Be careful, if you do not specify the -local flag, the command removes the Oracle Grid Infrastructure home from every node in the cluster.

- If you do not specify the -local flag, then the command removes the Oracle Grid Infrastructure home from every node in the cluster.
- If you cut and paste the preceding command, then paste it into a text editor before pasting it to the command line to remove any formatting this document might contain.

(Session 1)

```
[grid@host03 ~]$ /u01/app/12.2.0/grid/deinstall/deinstall -local
Checking for required files and bootstrapping ...
Please wait ...
...
Run the following command as the root user or the administrator on node
"host03".
/u01/app/12.2.0/grid/crs/install/rootcrs.sh -force -deconfig -paramfile
"/tmp/deinstall2017-12-21_06-46-15AM/response/deinstall_OraGI12Home1.rsp"
```

Press Enter after you finish running the above commands

<-----

(Session 2)

```
[root@host03 ~]# /u01/app/12.2.0/grid/crs/install/rootcrs.sh -force -
-deconfig -paramfile "/tmp/deinstall2017-12-21_06-46-
15AM/response/deinstall_OraGI12Home1.rsp"
...
2017/12/21 06:55:24 CLSRSC-336: Successfully deconfigured Oracle
Clusterware stack on this node
```

[root@host03 ~]#

(Session 1)

Press Enter after you finish running the above commands

<----- → Hit Enter

...

```
##### ORACLE DEINSTALL TOOL END #####
[grid@host03 ~]$
```

Step 4: As the root user, delete the node from the cluster from a node that will remain in the cluster.

```
[root@host01 ~]# crsctl delete node -n host03
CRS-4661: Node host03 successfully deleted.
[root@host01 ~]#
```

Deleting a Node from the Cluster

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

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

- If you remove a cluster node on which Oracle Clusterware is down, then determine whether the VIP for the deleted node still exists.

```
[grid@host01]$ srvctl config vip -node host03
```

- If the VIP still exist, then delete it.

```
[grid@host01]$ srvctl stop vip -node host03
```

```
[grid@host01]$ srvctl remove vip -node host03
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

[grid@host01 ~]\$ **cluvfy stage -post nodedel -n host03**

Verifying Node Removal ...

Verifying CRS Integrity ...PASSED

Verifying Clusterware Version Consistency ...PASSED

Verifying Node Removal ...PASSED

Post-check for node removal was successful.

CVU operation performed: stage -post nodedel

Date: Dec 21, 2017 7:23:03 AM

CVU home: /u01/app/12.2.0/grid/

User: grid

[grid@host01 ~]\$

Deleting a Node from a Windows-Based Cluster

To delete a node called `host03` from a Windows-based cluster:

1. From a local home on `host03`, run the following command:

```
C:\>Grid_home\oui\bin\setup.exe -updateNodeList ORACLE_HOME=Grid_home  
"CLUSTER_NODES={host03}" CRS=TRUE -local
```

2. Run the deinstall tool on the node you want to delete:

```
C:\Grid_home\deinstall\>deinstall.bat -local
```

3. Run this command from a node that you are not deleting:

```
C:\>Grid_home\bin\crsctl delete node -n host03
```

4. Verify that the node has been successfully deleted:

```
C:\>cluvfy stage -post nodedel -n node_list [-verbose]
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

To delete a node from a Windows-based cluster, perform the following steps:

1. Only if you have a local home, run the following command with the `-local` option to update the node list on the node you want to delete:

```
C:\>Grid_home\oui\bin\setup.exe -updateNodeList ORACLE_HOME=Grid_home  
"CLUSTER_NODES={node_to_be_deleted}" CRS=TRUE -local
```

2. Run the deinstall tool on the node you want to delete to deinstall and deconfigure the Oracle Clusterware home, as follows:

```
C:\Grid_home\deinstall\>deinstall.bat -local
```

3. On a node that you are not deleting, run the following command:

```
C:\>Grid_home\
```

4. Run the following CVU command to verify that the specified nodes have been successfully deleted from the cluster:

```
C:\>cluvfy stage -post nodedel -n node_list [-verbose]
```

Note: Oracle does not support using Oracle Enterprise Manager to delete nodes on Windows systems.

```
bin\crsctl delete node -n node_to_be_deleted
```



Quiz

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

- a. True
- b. False



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

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



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Practice 6: Overview

In this practice, you extend your cluster to another Hub Node and another Leaf Node.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Unauthorized reproduction or distribution prohibited. Copyright© 2019, Oracle and/or its affiliates.

GANG LIU (gangl@baylorhealth.edu) has a non-transferable license
to use this Student Guide.

Traditional Clusterware Management



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Demonstrate your Clusterware management proficiency
- Demonstrate Oracle Cluster Registry (OCR) backup and recovery techniques
- Manage network settings
- Describe the scope and capabilities of what-if command evaluation



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Managing Oracle Clusterware

Command-line utilities

- `crsctl` manages clusterware-related operations:
 - Starting and stopping non-Oracle based cluster resources
 - Enabling and disabling Oracle Clusterware daemons
 - Registering non-Oracle based cluster resources
- `srvctl` manages Oracle resource-related operations:
 - Starting and stopping database instances and services
- `oifcfg` can be used to define and administer network interfaces.
- `ocrconfig` can be used for OCR administration.
- `ocrcheck` and `ocrdump` are used to troubleshoot configuration problems that affect the OCR.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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. The `oifcfg` command-line interface helps you to define and administer network interfaces.

Oracle Cluster Registry Configuration Tool (`ocrconfig`) is a command-line tool for OCR administration. You can also use the `ocrcheck` and `ocrdump` utilities to troubleshoot configuration problems that affect OCR.

Role-Separated Management

- Role-separated management enables multiple applications and databases to share cluster and hardware resources.
 - This is done by setting permissions on server pools or resources to provide or restrict access to resources.
- Role-separated management can be implemented in one of two ways:
 - **Vertical implementation:** Based on different operating system users and groups used for various layers in the technology stack
 - **Horizontal implementation:** Restricts resource access within one layer using permissions for resources that are granted using access control lists assigned to server pools and policy-managed databases or applications



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Role-separated management is a feature you can implement that enables multiple applications and databases to share the same cluster and hardware resources, in a coordinated manner, by setting permissions on server pools or resources, in order to provide or restrict access to resources, as required. By default, this feature is not implemented during installation. You can implement role-separated management in one of two ways:

- Vertical implementation (between layers) describes a role separation approach based on different operating system users and groups used for various layers in the technology stack. Permissions on server pools and resources are granted to different users (and groups) for each layer in the stack using access control lists. Oracle Automatic Storage Management (ASM) offers setting up role separation as part of the Oracle Grid Infrastructure installation based on a granular assignment of operating system groups for specific roles.
- Horizontal implementation (within one layer) describes a role separation approach that restricts resource access within one layer using access permissions for resources that are granted using access control lists assigned to server pools and policy-managed databases or applications.

For example, consider an operating system user called `grid`, with primary operating system group `oinstall`, that installs Oracle Grid Infrastructure and creates two database server pools. The operating system users `ouser1` and `ouser2` must be able to operate within a server pool, but should not be able to modify those server pools so that hardware resources can be withdrawn from other server pools either accidentally or intentionally.

You can configure server pools before you deploy database software and databases by configuring a respective policy set.

Role-separated management in Oracle Clusterware no longer depends on a cluster administrator (but backward compatibility is maintained). By default, the user that installed Oracle Clusterware in the Oracle Grid Infrastructure home (Grid home) and `root` are permanent cluster administrators. Primary group privileges (`oinstall` by default) enable database administrators to create databases in newly created server pools using the Database Configuration Assistant, but do not enable role separation.

Note: Oracle recommends that you enable role separation before you create the first server pool in the cluster. Create and manage server pools by using configuration policies and a respective policy set. Access permissions are stored for each server pool in the ACL attribute.

Configuring Horizontal Role Separation

- **crsctl setperm** is used to configure horizontal role separation, assigning ACLs to server pools and resources.
- The syntax for the **crsctl setperm** is as follows:

```
# crsctl setperm {resource | type | serverpool} name {-u acl_string | -x
acl_string | -o user_name | -g group_name}
```

- To set permissions on a server pool called **psft** for the group **personnel**, where the administrative user has read/write/execute privileges, the members of the personnel group have read/write privileges, and users outside of the group are granted no access:

```
# crsctl setperm serverpool psft -u
user:personadmin:rwx,group:personnel:rw-,other::r--
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Use the **crsctl setperm** command to configure horizontal role separation by using ACLs that are assigned to server pools, resources, or both. The **crsctl** utility is located in the path *Grid_home/bin*, where *Grid_home* is the Oracle Grid Infrastructure for a cluster home. The command uses the following syntax, where the access control (ACL) string is indicated by *italics*:

```
crsctl setperm {resource | type | serverpool} name {-u acl_string | -x
acl_string | -o user_name | -g group_name}
```

The flag options are:

- u**: Update the entity ACL
- x**: Delete the entity ACL
- o**: Change the entity owner
- g**: Change the entity primary group

The ACL strings are:

```
{ user:user_name[:readPermwritePermexecPerm] |
group:group_name[:readPermwritePermexecPerm] |
other[::readPermwritePermexecPerm] }
```

where:

user: Designates the user ACL (access permissions granted to the designated user)

group: Designates the group ACL (permissions granted to the designated group members)

other: Designates the other ACL (access granted to users or groups not granted particular access permissions)

readperm: Location of the read permission (**r** grants permission and **-** forbids permission)

writeperm: Location of the write permission (**w** grants permission and **-** forbids permission)

execperm: Location of the execute permission (**x** grants permission and **-** forbids permission)

Controlling Oracle Clusterware

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

- To start or stop Oracle Clusterware on the local server:

```
# crsctl start cluster
```

```
# crsctl stop cluster
```

- To start or stop OHAS on the local server:

```
# crsctl start crs
```

```
# crsctl stop crs
```

- To enable or disable Clusterware on the local server:

```
# crsctl enable crs
```

```
# crsctl disable crs
```

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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 cluster
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 Clusterware by using the `crsctl stop cluster` command and start it by using the `crsctl start cluster` command. If you do not specify `-all` or one or more space-delimited server names, Oracle Clusterware stops the Oracle Clusterware stack on the local server. If any resources that Oracle Clusterware manages are still running after you run the `crsctl stop cluster` command, the command fails. Use the `-n` option to specify a list of nodes. Use the `-f` option to unconditionally stop all resources and stop the Clusterware stack.

Use the `crsctl config crs` command to display Oracle High Availability Services automatic startup configuration.

```
# crsctl config crs
CRS-4622: Oracle High Availability Services autostart is enabled.
```

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) on a specific node:

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



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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 -all` command:

```
$ crsctl check cluster -all
*****
host01:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
*****
host02:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
*****
```

Determining the Location of Oracle Clusterware Configuration Files

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

- To determine the location of the OCR:

```
$ ocrcheck -config  
Oracle Cluster Registry configuration is :  
Device/File Name : +DATA
```

- To determine the location of the voting disk:

```
$ crsctl query css votedisk  
## STATE File Universal Id File Name Disk group  
-- ----- -----  
1. ONLINE 620281e2ca184ffbbf549a3ea0326ecf (/dev/c1_DATA1_dsk13) [DATA]  
2. ONLINE 4222108c7a504f0abf464cb75bb555e6 (/dev/c1_DATA1_dsk2) [DATA]  
3. ONLINE ff5d5cccd2fa04f70bffba241b89de6be (/dev/c1_DATA1_dsk14) [DATA]  
Located 3 voting disk(s).
```

ORACLE

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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.

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.

```
# ocrcheck  
Status of Oracle Cluster Registry is as follows :  
Version : 4  
Total space (kbytes) : 409568  
Used space (kbytes) : 5040  
Available space (kbytes) : 404528  
ID : 2020194090  
Device/File Name : +DATA  
Device/File integrity check succeeded  
...
```

Checking the Integrity of Oracle Clusterware Configuration Files

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

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

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

```
# ocrcheck
Status of Oracle Cluster Registry is as follows :
  Version          :        4
  Total space (kbytes)   :  409568
  Used space (kbytes)    :   3320
  Available space (kbytes) : 406248
  ID                 : 92167993
  Device/File Name     : +DATA
                         Device/File integrity check succeeded
...
Cluster registry integrity check succeeded
Logical corruption check succeeded
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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
2015-06-16 14:59:17.246722 :      CSSD:3175163648: clssnmCompleteVFDDiscovery:
    Completing voting file discovery
2015-06-16 14:59:17.246726 :      CSSD:3175163648:
    clssnmvVerifyCommittedConfigVFs: Insufficient voting files found, found 0
    of 0 configured, needed 1 voting files
```

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

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

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
host01 2018/01/19 17:42:16 +MGMT:/cluster01/OCRBACKUP/backup00.ocr.285.965842927 0
host01 2018/01/19 13:42:06 +MGMT:/cluster01/OCRBACKUP/backup01.ocr.291.965828517 0
host01 2018/01/19 09:41:55 +MGMT:/cluster01/OCRBACKUP/backup02.ocr.281.965814105 0
host01 2018/01/18 05:40:38 +MGMT:/cluster01/OCRBACKUP/day.ocr.286.965713239 0
host01 2018/01/11 02:00:04 +MGMT:/cluster01/OCRBACKUP/week.ocr.283.965095205 0
```

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



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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.

As of Oracle Clusterware 12c release 2 (12.2), the default location for generating backups is an Oracle ASM disk group, which you can change between Oracle ASM disk groups, but you *cannot* change to a local file system.

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. At the end of the week, a backup is taken. 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 file_name
```

- 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.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Considerations of Managing OCR and Voting Disks

- Oracle Clusterware 12c does not support the use of raw or block devices. To upgrade to Oracle Clusterware 12c from a previous Oracle Clusterware release on which you were using raw or block devices, you must migrate OCR and voting files to Oracle ASM or a shared file system before you upgrade.
- Beginning with Oracle Clusterware 12c release 2 (12.2), the placement of OCR and voting files directly on a shared file system is deprecated in favor of having those files managed by Oracle ASM.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware 12c does not support the use of raw or block devices. To upgrade to Oracle Clusterware 12c from a previous Oracle Clusterware release on which you were using raw or block devices, you must migrate OCR and voting files to Oracle ASM or a shared file system before you upgrade.

Beginning with Oracle Clusterware 12c release 2 (12.2), the placement of OCR and voting files directly on a shared file system is deprecated in favor of having those files managed by Oracle ASM.

If you must use a supported shared file system (either a network file system or a shared cluster file system) instead of native disk devices, then you must create Oracle ASM disks on the shared file system that you plan to use for hosting OCR and the voting files before installing Oracle Grid Infrastructure. You can then use the Oracle ASM disks in an Oracle ASM disk group to manage OCR and the voting files.

This lesson will show how to manage Oracle Cluster Registry and Voting Disk Files in both Pre-12c Release 2 and 12c Release 2.

Adding, Replacing, and Repairing OCR Locations

- To 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:

```
# ocrconfig -repair -add +DATA1
```

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



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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.

Note: You cannot repair the OCR configuration on a node on which the Oracle Cluster Ready Services daemon is running.

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/sd1
```

Note: Do not perform an OCR removal unless there is at least one other active OCR location online. Otherwise, you get an error.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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 because this will result in an error. You cannot remove the last OCR file.

Migrating OCR Locations to ASM

1. Ensure that the upgrade to Oracle Clusterware 12c is complete.

```
$ crsctl query crs activeversion  
Oracle Clusterware active version on the cluster is [12.2.0.1.0]
```

2. Start ASM on all nodes and 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, run the following command as `root`:

```
# ocrconfig -add +DATA2
```

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

```
# ocrconfig -delete /dev/raw/raw1  
# ocrconfig -delete /dev/raw/raw2
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

To improve Oracle Clusterware storage manageability, OCR is configured, by default, to use ASM in Oracle Database 12c. With the Clusterware storage residing in an ASM disk group, you can manage both database and clusterware storage by using Oracle Enterprise Manager.

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`, perform the following steps:

1. Ensure that the upgrade of Oracle Clusterware to 12c 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 the upgrade to Oracle Clusterware 12c is complete.

```
$ crsctl query crs activeversion  
Oracle Clusterware active version on the cluster is [12.2.0.1.0]
```

2. Create at least one shared file with the following permissions: root, oinstall, 640. Ensure that the mount partition has at least 400 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 /nas/ocr
```

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

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



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

To migrate Oracle Clusterware storage from ASM to another storage choice, perform the following steps:

1. Ensure that the upgrade to Oracle Clusterware 12c 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 400 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

When significant changes to the configuration have occurred, a manual on-demand backup is suggested.

- To perform a physical backup:

```
# ocrconfig -manualbackup
```

- To display a list of manual backups:

```
$ ocrconfig -showbackup manual
host01    2018/01/16 18:07:34
+MGMT:/cluster01/OCRBACKUP/backup_20180116_180734.ocr.289.965585255      0
host01    2018/01/12 16:32:34
+MGMT:/cluster01/OCRBACKUP/backup_20180112_163234.ocr.288.965233955      0
host02    2018/01/12 14:35:30
+MGMT:/cluster01/OCRBACKUP/backup_20180112_143530.ocr.284.965226931      0
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Restoring the OCR on Linux or UNIX Systems

Case 1: OCR in a Non-ASM Storage

1. List the cluster nodes by running the `olsnodes` command.
2. Stop Oracle Clusterware by running the following command as `root` on all of the nodes:

```
# crsctl stop crs [-f]
```

3. If restoring to a cluster or network file system, run the following command as `root` to restore the OCR:

```
# ocrconfig -restore /u01/app/.../cdata/cluster01/day.ocr
```

4. Run this command on all nodes to start Clusterware:

```
# crsctl start crs
```

5. Verify the OCR integrity of all cluster nodes:

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



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

If you are storing OCR on an Oracle ASM disk group, and that disk group is corrupt, you must restore the Oracle ASM disk group by using Oracle ASM utilities, and then mount the disk group again before recovering OCR. Recover OCR by running the `ocrconfig -restore` command, as instructed in the following procedure.

1. List the nodes in your cluster by running the following command on one node:
`$ olsnodes`
2. Stop Oracle Clusterware by running the following command as `root` on all of the nodes:
`# crsctl stop crs [-f]`
3. If you are restoring OCR to a cluster or network file system, run the following command to restore OCR with a backup identified by using the `ocrconfig -showbackup` command.
`# ocrconfig -restore file_name`
4. Begin to start Clusterware by running the following command as `root` on all of the nodes:
`# crsctl start crs`
5. Verify OCR integrity of all of the cluster nodes by running the following CVU command:
`$ cluvfy comp ocr -n all -verbose`

Restoring the OCR on Linux or UNIX Systems

Case 2: OCR in an ASM Disk Group

1. List the cluster nodes by running the `olsnodes` command.
2. Stop Oracle Clusterware by running the following command as `root` on all of the nodes:

```
# crsctl stop crs [-f]
```
3. Run the following command to start the Clusterware stack on one node in exclusive mode:

```
# crsctl start crs -excl -nocrs
```
4. If you are restoring to an Oracle ASM disk group, create a disk group that has the same name as the disk group that you want to restore.
5. Restore OCR with a backup that you can identify.

```
# ocrconfig -restore +MGMT:/cluster01/OCRBACKUP/day.ocr.282.965626773
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

If you are storing OCR on an Oracle ASM disk group, and that disk group is corrupt, you must restore the Oracle ASM disk group by using Oracle ASM utilities, and then mount the disk group again before recovering OCR. Recover OCR by running the `ocrconfig -restore` command, as instructed in the following procedure.

1. List the nodes in your cluster by running the following command on one node:
`$ olsnodes`
2. Stop Oracle Clusterware by running the following command as `root` on all of the nodes:
`# crsctl stop crs [-f]`
3. Run the following command to start the Clusterware stack on one node in exclusive mode:
`# crsctl start crs -excl -nocrs`

The `-nocrs` option ensures that the CRSD process and OCR do not start with the rest of the Oracle Clusterware stack. Ignore any errors that display. Check whether CRSD is running.

If it is, stop it by running the following command as `root`:

- `# crsctl stop resource ora.crsd -init`

The `-nocrs` option ensures that the CRSD process and OCR do not start with the rest of the Oracle Clusterware stack. Ignore any errors that display. Check whether CRSD is running. If it is, stop it by running the following command as `root`:

```
# crsctl stop resource ora.crsd -init
```

4. If you want to restore OCR to an Oracle ASM disk group, you must first create a disk group by using SQL*Plus that has the same name as the disk group you want to restore and mount it on the local node. If you cannot mount the disk group locally, run the following SQL*Plus command:
SQL> drop diskgroup disk_group_name force including contents;
5. Restore OCR with a backup that you can identify by running the following command as root:
ocrconfig -restore file_name

Restoring the OCR on Linux or UNIX Systems

6. Verify its integrity as shown in the following:

```
# ocrcheck
```

7. Stop Clusterware where it is running in exclusive mode:

```
# crsctl stop crs -f
```

8. Run `ocrconfig -repair` on all nodes where you did *not* run the `ocrconfig -restore` command:

```
# ocrconfig -repair -replace
```

9. Run this command on all nodes to start Clusterware:

```
# crsctl start crs
```

10. Verify the OCR integrity of all cluster nodes:

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



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

6. Verify the integrity of OCR:

```
# ocrcheck
```

7. Stop Oracle Clusterware on the node where it is running in exclusive mode:

```
# crsctl stop crs -f
```

8. Run the `ocrconfig -repair -replace` command as `root` on all the nodes in the cluster where you did not run the `ocrconfig -restore` command. For example, if you ran the `ocrconfig -restore` command on node 1 of a four-node cluster, you must run the `ocrconfig -repair -replace` command on nodes 2, 3, and 4.

9. Begin to start Clusterware by running the following command as `root` on all of the nodes:

```
# crsctl start crs
```

10. Verify OCR integrity of all of the cluster nodes by running the following CVU command:

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

Backing Up and Recovering the Voting Disk

- In Oracle Clusterware 12c, 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.
- To add or remove voting disks on non–Automatic Storage Management (ASM) storage, use the following commands:

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



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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.

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 12.1.0.0.

Adding, Deleting, or Migrating Voting Disks

- To add or delete one or more voting disks to non-ASM storage:

```
# crsctl add css votedisk <path_to_new_voting_disk>
# crsctl delete css votedisk <path_to_old_voting_disk>
```

- To move a voting disk to an ASM disk group:

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

- 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:

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



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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 move a voting disk to an ASM disk group:

```
# 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 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.

Restoring Voting Disks

To restore voting disks if all of them are corrupted:

1. Restore the OCR, if necessary.
2. Run the following command from only one node to start the Clusterware stack in exclusive mode:

```
# crsctl start crs -excl
```

3. Run the `crsctl query css votedisk` command to retrieve the list of currently defined voting files:

```
$ crsctl query css votedisk
## STATE File Universal Id          File Name Disk group
-- -----
1. ONLINE 620281e2ca184ffbbf549a3ea0326ecf (/dev/c1_DATA1_dsk13) [DATA]
2. ONLINE 4222108c7a504f0abf464cb75bb555e6 (/dev/c1_DATA1_dsk2) [DATA]
3. ONLINE ff5d5cccd2fa04f70bffba241b89de6be (/dev/c1_DATA1_dsk14) [DATA]
Located 3 voting disk(s).
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

If all of the voting disks are corrupted, you can restore them, as follows:

1. Restore OCR as described earlier in this lesson, if necessary. This step is necessary only if OCR is also corrupted or otherwise unavailable, such as if OCR is on Oracle ASM and the disk group is no longer available.
2. Run the following command from only one node to start the Oracle Clusterware stack in exclusive mode, which does not require voting files to be present or usable:

```
# crsctl start crs -excl
```

3. Run the `crsctl query css votedisk` command to retrieve the list of voting files currently defined, similar to the following:

```
$ crsctl query css votedisk
## STATE File Universal Id          File Name Disk group
-- -----
1. ONLINE 620281e2ca184ffbbf549a3e0326ecf (/dev/c1_DATA1_dsk13) [DATA]
2. ONLINE 4222108c7a504f0abf464cb75bb555e6 (/dev/c1_DATA1_dsk2) [DATA]
3. ONLINE ff5d5cccd2fa04f70bffba21b89de6be (/dev/c1_DATA1_dsk14) [DATA]
Located 3 voting disk(s).
```

This list may be empty if all voting disks were corrupted, or may have entries that are marked as status 3 or OFF.

Restoring Voting Disks

Case 1: Voting Disks in a Non-ASM Storage

4. If you did not store voting disks in ASM, delete the voting disk using the FUID obtained in the previous step:

```
$ crsctl delete css votedisk <File Universal ID>
```

5. Add a voting disk:

```
$ crsctl add css votedisk <path_to_voting_disk>
```

6. Stop the Oracle Clusterware stack:

```
# crsctl stop crs -f
```

7. Restart the Clusterware stack in normal mode:

```
# crsctl start crs
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

4. If you did not store voting disks in Oracle ASM, run the following command using the File Universal Identifier (FUID) obtained in the previous step:

```
$ crsctl delete css votedisk FUID
```

5. Add a voting disk, as follows:

```
$ crsctl add css votedisk path_to_voting_disk
```

6. Stop the Oracle Clusterware stack as the root user:

```
# crsctl stop crs -f
```

7. Restart the Oracle Clusterware stack in normal mode as the root user:

```
# crsctl start crs
```

Restoring Voting Disks

Case 2: Voting Disks in an ASM Disk Group

4. If the voting disks are stored in ASM, migrate the voting disks to the Oracle ASM disk group that you specify:

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

5. Stop the Oracle Clusterware stack:

```
# crsctl stop crs -f
```

6. Restart the Clusterware stack in normal mode:

```
# crsctl start crs
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

4. If the voting disks are stored in Oracle ASM, run the following command to migrate the voting disks to the Oracle ASM disk group you specify:

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

The Oracle ASM disk group to which you migrate the voting files must exist in Oracle ASM.

You can use this command whether the voting disks were stored in Oracle ASM or some other storage device.

5. Stop the Oracle Clusterware stack as the `root` user:

```
# crsctl stop crs -f
```

6. Restart the Oracle Clusterware stack in normal mode as the `root` user:

```
# crsctl start crs
```

Oracle Local Registry

- For node-specific resources, each cluster node has a local registry 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/12.1.0/grid/cdata/host01.olr
  Device/File integrity check succeeded
  Local registry integrity check succeeded
  Logical corruption check succeeded
```

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Each node in a cluster has a local registry for node-specific resources, called an Oracle Local Registry (OLR), which 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 by using the `ocrcheck`, `ocrdump`, and `ocrconfig` utilities with the `-local` option. To see the location of the OLR, use the `ocrcheck` utility:

```
# ocrcheck -config -local
```

You can check the status of the OLR as follows:

```
# ocrcheck -local
```

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.

Oracle Interface Configuration Tool: `oifcfg`

- The `oifcfg` command-line interface helps you to define and administer network interfaces.
- The `oifcfg` is a command-line tool for both single-instance Oracle databases and Oracle RAC environments.
- The `oifcfg` utility can direct components to use specific network interfaces.
- The `oifcfg` utility can be used to retrieve component configuration information.

```
$ oifcfg getif
eth0 192.0.2.0 global public
eth2 192.168.1.0 global cluster_interconnect
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The Oracle Interface Configuration Tool (`oifcfg`) command-line interface helps you to define and administer network interfaces. The Oracle Interface Configuration Tool (`oifcfg`) is a command-line tool for both single-instance Oracle databases and Oracle RAC environments. You can use `oifcfg` commands in Oracle Clusterware environments to:

- Allocate and deallocate network interfaces to components
- Direct components to use specific network interfaces
- Retrieve component configuration information

Before you invoke `oifcfg`, ensure that you have started Oracle Clusterware on at least the local node and preferably on all nodes if you intend to include the `-global` option on the command.

Run `oifcfg` from the `Grid_home/bin/` directory as the user who installed the Oracle Clusterware software.

Determining the Current Network Settings

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

```
$ oifcfg iflist -n
eth0 192.0.2.0 255.255.255.0
eth1 192.0.3.0 255.255.255.0
eth2 192.168.1.0 255.255.255.0
eth2 169.254.0.0 255.255.0.0
eth3 192.168.2.0 255.255.255.0
```

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

```
$ srvctl config nodeapps -a
Network 1 exists
Subnet IPv4: 192.0.2.0/255.255.255.0/eth0, dhcp
Subnet IPv6:
VIP exists: network number 1, hosting node host01
VIP IPv4 Address: -/host01-vip/192.0.2.237
VIP IPv6 Address:
VIP exists: network number 1, hosting node host02
VIP IPv4 Address: -/host02-vip/192.0.2.231
VIP IPv6 Address:
```

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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.

Configuring Redundant Interconnect Usage Using OIFCFG

- Redundant Interconnect Usage can be configured after Clusterware installation.
- Use the `oifcfg` command to designate unused network interfaces as private.

```
$ oifcfg setif -global \
eth3/192.168.2.0:cluster_interconnect
```

- Clusterware creates from one to four highly available IP (HAIP) addresses.
- After modifying the interfaces:
 - Stop Clusterware on all nodes.

```
# crsctl stop crs
```

- Start Clusterware on all nodes.

```
# crsctl start crs
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

You can define multiple interfaces for Redundant Interconnect Usage after Clusterware installation by classifying the interfaces as private by using the `oifcfg setif` command. When you do, Oracle Clusterware creates from one to four (depending on the number of interfaces you define) highly available IP (HAIP) addresses, which Oracle Database and Oracle ASM instances use to ensure highly available and load-balanced communications.

The Oracle software (including Oracle RAC, Oracle ASM, and Oracle ACFS, all 11.2.0.2, or later), by default, uses these HAIP addresses for all of its traffic, allowing for load balancing across the provided set of cluster interconnect interfaces. If one of the defined cluster interconnect interfaces fails or becomes noncommunicative, Oracle Clusterware transparently moves the corresponding HAIP address to one of the remaining functional interfaces.

After the network interfaces have been designated private by using the `oifcfg setif` command, Clusterware must be stopped on all nodes and restarted by using the `crsctl stop crs` and `crsctl start crs` commands, respectively.

Changing the Virtual IP Addresses Using SRVCTL

- Stop all services running on the node whose VIP address you want to change:

```
$ srvctl stop service -db orcl -service sales,oltp \
-node host01
```

- Confirm the current IP address for the VIP address:

```
$ srvctl config vip -node host01
VIP exists: network number 1, hosting node host01
VIP IPv4 Address: -/host01-vip/192.0.2.237
VIP IPv6 Address:
```

- Stop the VIP address:

```
$ srvctl stop vip -node host01
```

- Verify that the VIP address is no longer running by executing the ifconfig -a command.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Clients configured to use Public VIP addresses for Oracle Database releases before Oracle Database 11g 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:

- Stop all services running on the node whose VIP address you want to change:

```
$ srvctl stop service -db db_name -service service_name_list -node
my_node
```

- Confirm the current IP address for the VIP address by using the `srvctl config vip` command:

```
$ srvctl config vip -node my_node
```

- Stop the VIP address by using the `srvctl stop vip` command:

```
$ srvctl stop vip -node mynode
```

- Verify that the VIP address is no longer running by using the `ifconfig -a` command.

Changing the Virtual IP Addresses Using SRVCTL

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 -node host01 -address \
192.0.2.125/255.255.255.0/eth0
```

7. Start the node VIP.

```
# srvctl start vip -node 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

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



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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 by using the following syntax:

```
# srvctl modify nodeapps -node node_name -address new_vip_address
```

7. Start the node VIP by running the `srvctl start vip` command:

```
$ srvctl start vip -node 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
```

The example in the slides are to describe how to change only a VIP address, and assume that the host name associated with the VIP address does not change. If you are using GNS, and VIPs are assigned using DHCP, you do not need to update VIP addresses manually if you are using GNS, and VIPs are assigned using DHCP.

To use a different subnet or network interface card for the default network before you change any VIP resource, you must use the `srvctl modify network`

`subnet subnet/netmask/interface` command as root to change the network resource, where `subnet` is the new subnet address, `netmask` is the new netmask, and `interface` is the new interface. After you change the subnet, then you must change each node's VIP to an IP address on the new subnet, as described in step 6.

Changing the Interconnect Adapter Using OIFCFG

- On a single node in the cluster, add the new global interface specification:

```
$ oifcfg setif -global eth1/192.0.3.0:cluster_interconnect
```

- Verify the changes with `oifcfg getif` and then stop Clusterware on all nodes by running the following command as `root` on each node:

```
# oifcfg getif
# crsctl stop crs
```

- Assign the network address to the new network adapters on all nodes using `ifconfig`:

```
# ifconfig eth2 192.0.2.15 netmask 255.255.255.0 \
broadcast 192.0.2.255
```

- Remove the former adapter/subnet specification and restart Clusterware:

```
$ oifcfg delif -global eth2/192.168.1.0
# crsctl start crs
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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:

- Make sure that the Oracle Clusterware stack is up and running on all cluster nodes.
- 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 nodes.
- On a single node in the cluster, add the new global interface specification:
`$ oifcfg setif -global interface_name/subnet:cluster_interconnect`
- On a node in the cluster, use `ifconfig` to ensure that the new IP address exists.
- 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`
- 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
```

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.

9. 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.

10. 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

- To view SCAN VIP configuration:

```
# srvctl config scan
SCAN name: cluster01-scan.cluster01.example.com, Network: 1
Subnet IPv4: 192.0.2.0/255.255.255.0/eth0
SCAN 0 IPv4 VIP: -/scan1-vip/192.0.2.240
SCAN name: cluster01-scan.cluster01.example.com, Network: 1
Subnet IPv4: 192.0.2.0/255.255.255.0/eth0
SCAN 1 IPv4 VIP: -/scan2-vip/192.0.2.239
SCAN name: cluster01-scan.cluster01.example.com, Network: 1
Subnet IPv4: 192.0.2.0/255.255.255.0/eth0
SCAN 2 IPv4 VIP: -/scan3-vip/192.0.2.232
```

- To view SCAN Listener configuration:

```
# srvctl config scan_listener
SCAN Listener LISTENER_SCAN1 exists. Port: TCP:1521
Registration invited nodes:
Registration invited subnets:
SCAN Listener LISTENER_SCAN2 exists. Port: TCP:1521
Registration invited nodes:
Registration invited subnets:
SCAN Listener LISTENER_SCAN3 exists. Port: TCP:1521
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The `srvctl config scan` command is used to view the cluster's SCAN VIP configuration. The command displays the cluster SCAN name and the network IP and netmask. In addition, the command displays the SCAN VIP resource name and the virtual IP address for the SCAN VIP.

The `srvctl config scan_listener` command is used to display configuration information for the SCAN listeners. The command shows how many SCAN listeners are configured, their resource names, and the listener port.

These commands should be used to check the current SCAN VIP and SCAN listener configuration before making any changes.

Managing SCAN VIP and SCAN Listener Resources

- To add a SCAN VIP resource:

```
# srvctl add scan -scanname cluster01-scan.cluster01.example.com
```

- To remove Clusterware resources from SCAN VIPs:

```
# srvctl remove scan [-f]
```

- To add a SCAN listener resource:

```
# srvctl add scan_listener -listener myscanlistener
# srvctl add scan_listener -endpoints "TCP:65536"
## using nondefault port number ##
```

- To remove Clusterware resources from all SCAN listeners:

```
# srvctl remove scan_listener [-f]
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The `srvctl add scan` command adds Oracle Clusterware resources for the given SCAN. This command creates the same number of SCAN VIP resources as the number of IP addresses that SCAN resolves to, or 3 when `network_number` identifies a dynamic network and Oracle GNS configuration. For static networks, the addresses to which the SCAN resolves in DNS must match the address type of the subnet. For an IPv4 network, the SCAN must resolve to IPv4 addresses.

`srvctl add scan -scanname scan_name [-netnum network_number]`

where `-scanname` is the domain name-qualified SCAN name and `-netnum` is the optional network number from which SCAN VIPs are obtained. If you do not specify this parameter, the SCAN VIPs are obtained from the same default network from which the `nodeapps` VIP is obtained.

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

```
# srvctl add scan -scanname 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.

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 -scanname cluster01-scan
```

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

```
# srvctl modify scan_listener -update
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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 -scanname scan_name
```

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

```
# srvctl modify scan -scanname 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 {-update | -endpoints [TCP:]port[/IPC:key]  
[/NMP:pipe_name] [/TCPS:s_port] [/SDP:port] } [-invitednodes node_list]  
[-invitedsubnets subnet_list]
```

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

```
# srvctl modify scan_listener -endpoints "TCP:1532"
# srvctl modify scan_listener -update
# srvctl config scan_listener ### To verify changes ####
SCAN Listener LISTENER_SCAN1 exists. Port: TCP:1532
Registration invited nodes:
Registration invited subnets:
SCAN Listener LISTENER_SCAN2 exists. Port: TCP:1532
Registration invited nodes:
Registration invited subnets:
SCAN Listener LISTENER_SCAN3 exists. Port: TCP:1532
Registration invited nodes:
Registration invited subnets:
```

SCAN Listeners and Valid Node Checking

- Valid node checking is used to specify nodes and subnets from which the SCAN listener accepts registrations.
- Specify the nodes and subnet information with `srvctl`.
 - `srvctl` stores the node and subnet information in the SCAN listener resource profile.
 - The SCAN listener agent reads that information from the resource profile and writes it to the `listener.ora` file.
- RAC releases prior to Oracle RAC 11g release 2 do not use SCAN listeners.
 - To support service registration for these databases, the `valid_node_check_for_registration_alias` value for the local listener in Oracle RAC 12c is set to `SUBNET`.
- To change the valid node checking settings for the node listeners, edit the `listener.ora` file.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

You can use valid node checking to specify the nodes and subnets from which the SCAN listener accepts registrations. You can specify the nodes and subnet information by using the `srvctl` utility. The `srvctl` utility stores the node and subnet information in the SCAN listener resource profile. The SCAN listener agent reads that information from the resource profile and writes it to the `listener.ora` file.

For single-instance databases, the local listener accepts service registrations only from database instances on the local node. Oracle RAC releases prior to Oracle RAC 11g release 2 (11.2) do not use SCAN listeners, and attempt to register their services with the local listener and the listeners defined by the `REMOTE_LISTENERS` initialization parameter. To support service registration for these older database instances, the default value of `valid_node_check_for_registration_alias` for the local listener in Oracle RAC 12c is set to the value `SUBNET`, rather than to the local node. To change the valid node checking settings for the node listeners, edit the `listener.ora` file.

SCAN listeners must accept service registration from instances on remote nodes. For SCAN listeners, the value of `valid_node_check_for_registration_alias` is set to `SUBNET` in the `listener.ora` file so that the corresponding listener can accept service registrations that originate from the same subnet.

You can configure the listeners to accept service registrations from a different subnet. For example, you might want to configure this environment when SCAN listeners share with instances on different clusters and nodes on those clusters are on a different subnet.

What-If Command Evaluation

Oracle Clusterware 12c provides a set of commands to preview a cluster management operation.

- Analyzes the impact before performing the operation
- Facilitates smooth operation of the cluster; no surprises
- Supported events:
 - Resource Start
 - Resource Stop
 - Resource Relocate
 - Resource Modify
 - Resource Add
 - Resource Failure
 - Server Pool Addition
 - Server Pool Removal
 - Server Pool Modification
 - Server Addition
 - Server Relocate
 - Server Removal
 - Set Active Policy



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware 12c provides a set of commands to determine the impact of a cluster management operation before the operation is actually executed. This capability is known as what-if command evaluation. It helps administrators to smoothly maintain the cluster and minimizes the potential for surprises. The slide lists the events supported by what-if command evaluation.

What-if command evaluation is supported using the clusterware C API, the `crsctl eval` command, and the `svrctl` command with the `-eval` option.

The following slides provide further details on performing what-if command evaluation by using the `crsctl` and `svrctl` commands. For further details regarding the clusterware API, see the API header file at `Grid_home/crs/demo/clscrsx.h`.

Performing What-If Command Evaluation on Application Resources with CRSCTL

- Commands for application administrators:

```
$ crsctl eval { start | stop | relocate | modify | add | fail } resource  
...
```

- Example:

```
$ crsctl eval start resource my_resource -n my_server  
Stage Group 1:  
-----  
Stage Number Required Action  
-----  
1 Y Resource 'my dep res1' (1/1) will be in state  
[ONLINE] on server [my_server]  
Resource 'my dep res2'-(1/1) will be in state  
[ONLINE|INTERMEDIATE] on server [my_server]  
2 Y Resource 'my resource' (1/1) will be in state  
[ONLINE|INTERMEDIATE] on server [my_server]
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Using the `crsctl eval` command, application administrators can perform what-if command evaluation to test the effect of starting, stopping, relocating, modifying, or adding cluster resources. Administrators can also examine the effect of a potential resource failure. The slide outlines the available commands. For more information, refer to the *Oracle Clusterware Administration and Deployment Guide 12c Release 2 (12.2)*.

What-if command evaluation using the `crsctl eval ... resource` command, as shown in this slide, is recommended for use only in conjunction with user-defined application resources. For Oracle Clusterware resources (resources with the `ora.` name prefix), you should use the `srvctl predict` or `srvctl ... -eval` command.

The bottom of the slide contains example output for a what-if scenario showing the effect of starting `my_resource` on `my_server`. In this example, the application administrator can clearly see that starting `my_resource` on `my_server` requires `my_dep_res1` to be started first. Also, before starting `my_resource`, Oracle Clusterware attempts to start `my_dep_res2`; however, the output notes that `my_dep_res2` is not a mandatory dependent resource.

Performing What-If Command Evaluation on Oracle Clusterware Resources with CRSCTL

- Commands for cluster administrators:

```
$ crsctl eval { add | delete | modify } serverpool ...
$ crsctl eval { add | relocate | delete } server ...
$ crsctl eval activate policy ...
```

- Example:

```
$ crsctl eval delete server my_server -f
Stage Group 1:
-----
Stage Number    Required      Action
-----
1              Y           Server 'some_server' will be moved from pools
                           [Free] to pools [ora.my_pool]
                           Server 'my_server' will be removed from pools
                           [ora.my_pool]
...
6              Y           Resource 'my_resource' (1/1) will be in state
                           [ONLINE] on server [some_server]
```

ORACLE

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Cluster administrators can use the `crsctl eval` command to perform what-if command evaluation that tests the effect of:

- Adding, deleting, and modifying server pools
- Adding servers to and deleting servers from a server pool
- Relocating a server from one server pool to another
- Removing a server from the cluster
- Enabling a specific management policy

The slide outlines the available commands. For more information, refer to the *Oracle Clusterware Administration and Deployment Guide 12c Release 1 (12.1)*.

The example in the slide contains partial output for a what-if scenario showing the effect of removing `my_server` from the cluster. In this example, removing `my_server` causes `some_server` to be moved into the `my_pool` server pool. After the server pool reallocation, the required resources, including `my_resource` and its dependent resources, are started on `some_server`.

Formatting the Output for What-If Command Evaluation on Oracle Clusterware Resources

Commands for cluster administrators may contain additional parameters to govern the output format.

- Command syntax:

```
$ crsctl eval ... serverpool ... [-admin [-l <level>] [-x] [-a]]  
$ crsctl eval ... server ... [-admin [-l <level>] [-x] [-a]]  
$ crsctl eval activate policy ... [-admin [-l <level>] [-x] [-a]]
```

- Example:

```
$ crsctl eval ac$ crsctl eval delete server my_server -admin -a  
  
NAME = Free  
ACTIVE_SERVERS =  
  
NAME = Generic  
ACTIVE_SERVERS =  
  
NAME = ora.my_pool  
ACTIVE_SERVERS = some_server  
tivate policy ... [-admin [-l <level>] [-x] [-a]]
```

ORACLE

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The `crsctl eval` commands for cluster administrators may contain additional parameters to govern the format of the command output. Their aim is to provide cluster administrators with the ability to control the amount of information returned by the commands.

The slide contains an example using the `-admin` option by itself. Rather than showing output describing all of the effects on servers and resources (like the example on the previous page), the `-admin` option modifies the output so that the administrator is provided with a summary of the server pool assignments resulting from the proposed action.

Following is a brief description of the additional formatting parameters that administrators can use in conjunction with the `-admin` option:

- l <level> specifies the output display level.
 - l serverpools displays server pool information.
 - l resources displays resource information.
 - l all displays server pool and resource information.
- x shows differences only.
- a shows all resources.

Performing What-If Command Evaluation with SRVCTL

- Commands:

```
$ srvctl { add | start | stop | modify | relocate } database ... -eval  
$ srvctl { add | start | stop | modify | relocate } service ... -eval  
$ srvctl { add | modify | remove } svrpool ... -eval  
$ srvctl relocate server ... -eval
```

- Example:

```
$ srvctl start database -db orcl -eval  
Resource ora.asm will be started on node c00n02  
Resource ora.DATA.dg will be started on node c00n02  
Resource ora.FRA.dg will be started on node c00n02  
Database orcl will be started on node c00n02
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

In addition to the `crsctl eval` commands that perform what-if command evaluation, administrators can use the `srvctl` command with the `-eval` option to perform what-if command evaluation that tests the effect of:

- Adding, starting, stopping, modifying, and relocating databases
- Adding, starting, stopping, modifying, and relocating services
- Adding, modifying, and removing server pools
- Relocating a server from one server pool to another

The slide outlines the available commands. For more information, refer to the *Oracle Real Application Clusters Administration and Deployment Guide 12c Release 2 (12.2)*.

The example in the slide contains output for a what-if scenario showing the effect of starting a database. In this example, the database administrator can clearly see that starting the database also causes ASM and the required disk group resources to start.

Evaluating Failure Consequences with SRVCTL

- Command:

```
$ srvctl predict { database | service | asm | diskgroup | filesystem |
                   vip | network | listener | scan | scan_listener |
                   oc4j } ... [-verbose]
```

- Examples:

```
$ srvctl predict asm -n c00n02
```

```
$ srvctl predict diskgroup -g DATA
```

```
$ srvctl predict filesystem -d /dev/asm/vol1-261 -verbose
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The `srvctl predict` command allows administrators to evaluate the consequences of a failure affecting any of the following types of resources:

- Database
- Service
- ASM
- Diskgroup
- Filesystem
- VIP
- Network
- Listener
- SCAN
- SCAN listener
- OC4J

For more information regarding the `srvctl predict` command options, refer to the *Oracle Real Application Clusters Administration and Deployment Guide 12c Release 2 (12.2)*.

Reasoned Command Evaluation (Why-If)

- Reasoned What-If command evaluation provides the rationale behind the policy decisions.
- It explains the entities involved, their attributes, and the criteria used to arrive at each of the potential actions.
- You can use `crsctl` to evaluate *what* will happen using the `eval` argument and *why* using the `-explain` argument.
- Reasoned command evaluation allows you to manage servers, server pools, and policies without making any actual changes.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware 12c Release 1 provided a set of evaluation commands and APIs to determine the impact of a certain operation before executing the operation. With the introduction of Oracle Clusterware 12c Release 2, the reasoned What-If command evaluation feature provides the rationale behind the policy decisions and explains the entities involved, their attributes, and the criteria used to arrive at each of the potential actions. You can use the `crsctl` utility to evaluate what will happen by using the `eval` argument and why it happened using the `-explain` argument. The reasoned command evaluation feature of the `crsctl` command allows you to manage the servers, server pools, and policies within your Clusterware environment without making any actual changes.

The Why-If command evaluations help the applications, cluster, and system administrators who are involved in capacity planning and configuration management to set up and test resource management policies.

Why-If: Managing Servers, Server Pools, and Policies

Reasoned command evaluation allows you to manage servers, server pools, and policies without making any actual changes.

```
$ crsctl eval add server host03 -explain

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

1   E      Looking for a suitable server pool for server 'host03'
       E      Scanning server pools with MIN_SIZE or more servers in
              descending order of IMPORTANCE.
       E      Considering server pools (IMPORTANCE): Free(0) for relocating servers.
       E      Relocating server 'host03' to server pool 'Free'.
       Y      Server 'host03' will be added to pools 'Free'.
2   Y      Resource 'ora.LISTENER_SCAN2.lsnr'(1/1) will be in state
              'OFFLINE'.
       N      Resource 'ora.ASMNET1LSNR_ASM.lsnr' (host03) will be in
              state
              'ONLINE' on server 'host03'.
...
...
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The reasoned command evaluation feature of the `crsctl` command allows you to manage the servers, server pools, and policies within your Clusterware environment without making any actual changes. The Why-If command evaluations help the applications, cluster, and system administrators who are involved in capacity planning and configuration management to set up and test resource management policies.

```
$ crsctl eval activate policy p2 -explain
Stage Group 1:
-----
Stage Required Action

1   E      Starting to evaluate activation of policy 'p2' with server pools
              (MIN_SIZE, MAX_SIZE) 'Free(0,-1),sp1(2,-1),sp2(4,-1)'.
       E      Looking at other server pools to see whether MIN_SIZE value 4 of
              server pool 'sp2' can be met.
...
       E      Considering server pool 'sp1' because its MIN_SIZE is 2 and it
              has 0 servers above MIN_SIZE.
       E      Relocating server 'host01' to server pool 'sp2'.
       E      Scanning server pools with MIN_SIZE or fewer servers in
              ascending order of IMPORTANCE.
2   Y      Resource 'cs2' (3/1) will be in state 'ONLINE|INTERMEDIATE' on
              server 'host01'.
...
...
```

Quiz



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

- a. Enterprise Manager
- b. srvctl
- c. Oracle Universal Installer
- d. crsctl



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.



Quiz

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

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



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.



Quiz

Which command is recommended to evaluate the effect of ASM failure on a server?

- a. crsctl eval fail resource "ora.<node>.ASM<n>.asm"
- b. srvctl predict asm -n <node>
- c. Either of these commands



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Demonstrate your Clusterware management proficiency
- Demonstrate OCR backup and recovery techniques
- Manage network settings
- Describe the scope and capabilities of what-if command evaluation



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Practice 7: 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
- Using `oifcfg` to configure a second private interconnect
- Working with SCANs, SCAN listeners, and the GNS VIP
- Recovering from a voting disk corruption



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Policy-Based Cluster and Capacity Management



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe the architecture and components of policy-based cluster management
- Administer server categorization
- Administer a policy set
- Activate a policy
- Implement load-ware resource placement
- Implement server weight-based node eviction



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Policy-Based Cluster Management Enhancements: Overview

In previous releases:

- A cluster can be logically divided into server pools
 - The server pools collectively define a policy.
 - All nodes are assumed to be equal.
- “Quality of Service Management” uses a different policy
 - Potential for overlap, confusion, and inconsistency

With Oracle Clusterware 12c, policy-based cluster management is enhanced to provide:

- Extended server attributes to govern node placement
- A library of policy definitions with an easy way of switching between policies
- Unification of policies for Oracle Clusterware and “Quality of Service Management”



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware 11g, release 2 introduced policy-based cluster management. With this capability, a cluster can be logically divided into groups of servers known as server pools. The placement of nodes in each server pool is governed by the relative importance assigned to each server pool, along with other attributes such as the minimum and maximum number of nodes assigned to each server pool.

The server pool definitions effectively define a policy, which enables dynamic capacity assignment and fast resource failover when the number of nodes in the cluster changes. With release 11.2, all nodes are assumed to be equal; that is, there is no way to specify server attributes that favor server placement in a particular server pool.

With release 11.2, the Quality of Service (QoS) Management feature defines a separate policy for cluster resource management, which can be confusing for administrators unless they are familiar with all the policies. In addition, there exists a potential to create policies that are contrary to each other.

With Oracle Clusterware 12c, policy-based cluster management is enhanced in three important ways. First, numerous server attributes allow for greater flexibility and control over node assignments to different server pools. Second, an extended policy framework allows administrators to maintain a library of policies and easily switch between them as required. Finally, policy-based cluster management has been unified with QoS Management.

Server Pools

Server pools:

- Are logical divisions of a cluster into pools of servers/nodes
- Distribute a uniform workload over several servers in the cluster
- Are allocated to host databases or other resources
- Are managed by using the `crsctl` and `srvctl` commands
- Support parent/child relationships among server pools
 - Top-level pools are mutually exclusive.
- Include two built-in server pools at Oracle Clusterware installation:
 - **FREE**: For servers that are not assigned to other pools
 - **GENERIC**: For administrator-managed fixed configuration and for databases prior to **11g Release 2**



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Server pools are logical divisions of a cluster into pools of servers or nodes. They usually represent a subset of the total number of nodes a cluster contains. Server pools are allocated to host databases or other resources and determine on which nodes those resources may execute. Each server pool name must be unique within the cluster. Server pools distribute a uniform workload (set of Oracle Clusterware resources) over several servers in the cluster. They are managed by using the `crsctl` and `srvctl` commands. With role-separated management, you can explicitly grant permission to operating system users to change attributes of certain server pools. Server pools support parent/child relationships among the pools. The top-level server pools are always mutually exclusive, meaning that one server in the cluster can reside only in one particular server pool at a certain point in time. Top-level server pools create a logical division of the cluster into subclusters.

When Oracle Clusterware is installed, two server pools are created automatically: **GENERIC** and **FREE**. All servers in a new installation are assigned to the **FREE** server pool, initially. Servers move from **FREE** to newly defined server pools automatically.

Server Pools and Policy-Based Management

- With policy-based management, administrators specify the server pool in which the servers run.
 - A DBA uses `srvctl` to create a server pool for servers hosting a database or database service.
 - A clusterware administrator uses `crsctl` to create a server pool to host applications and other nondatabase uses.
- Server pools provide resource isolation to prevent applications running in one pool from accessing resources running in another pool.
- Oracle Clusterware provides fine-grained role separation between server pools.
 - This maintains required role separation in organizations that have clustered environments managed by separate groups.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

With policy-based management, administrators specify the server pool (excluding the Generic and Free pools) in which the servers run. For example, a database administrator uses `srvctl` to create a server pool for servers hosting a database or database service. A clusterware administrator uses `crsctl` to create server pools for nondatabase use, such as creating a server pool for servers hosting an application. Policy-based management:

- Enables online server reallocation based on a defined policy to satisfy workload capacity requirements
- Guarantees the allocation of required resources for critical work as defined by the policy
- Ensures isolation where necessary, so that you can provide dedicated servers in a cluster for applications and databases
- Enables policies to be configured to change pools in accordance with business needs or application demand, so that pools provide the required capacity at the right time

Server pools provide resource isolation to prevent applications running in one server pool from accessing resources running in another server pool. Oracle Clusterware provides fine-grained role separation between server pools. This capability maintains required management role separation between these groups in organizations that have clustered environments managed by separate groups.

Server Pool Attributes

| Attribute | Description |
|------------------------|---|
| ACL | Access Control List that defines privileges for the server pool |
| ACTIVE_SERVERS | List of servers currently assigned to a server pool |
| EXCLUSIVE_POOLS | Governs whether servers can be shared among other pools |
| IMPORTANCE | Relative importance of server pool ranging from 0 (lowest) to 1000 |
| MAX_SIZE | Maximum number of servers a server pool can contain |
| MIN_SIZE | Minimum number of servers a server pool can contain |
| NAME | The name of the server pool |
| PARENT_POOLS | Specifies parent pools when creating nested server pools |
| SERVER_CATEGORY | The name of a registered server category, used as part of server categorization |
| SERVER_NAMES | List of servers that may be associated with a server pool |



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Server pool attributes can be specified when you create a server pool, or they can be modified after a server pool has already been created. The only required attribute is **NAME**. The optional attributes include:

- **ACL:** Defines the owner of the server pool and what privileges are granted to various operating system users and groups. The value of this attribute is populated at the time a server pool is created based on the identity of the process creating the server pool, unless explicitly overridden. It uses a string in the format of: `owner:user:rwx,pgrp:group:rwx,other::r-` to specify allowed privileges for the owner, group, and other users. The privileges allowed for a server pool are read-only (`r`), the ability to modify attributes or delete the server pool (`w`), and the ability to assign resources to the pool (`x`).
- **ACTIVE_SERVERS:** Is a space-delimited list of servers that are currently assigned to the server pool. This attribute is automatically managed by Oracle Clusterware.
- **EXCLUSIVE_POOLS:** Governs whether servers assigned to this server pool are shared with other server pools. This attribute is a string value containing any arbitrary string. Any other server pool that has the same value for this string is mutually exclusive with this server pool.

- **IMPORTANCE:** Determines the relative importance of the server pool compared to other server pools, with 0 denoting the lowest level of importance and 1000 the highest level of importance. The default value is 0.
- **MAX_SIZE:** Determines the maximum number of servers a server pool can contain. A value of –1 for this attribute spans the entire cluster and is the default value.
- **MIN_SIZE:** Determines the minimum number of servers a server pool can contain. The value of this attribute does not set a hard limit. It governs the priority for server assignment. The default value is 0.
- **PARENT_POOLS:** Allows the creating of nested server pools. Server pools listed in this attribute are referred to as parent server pools. Multiple parent server pools may be specified by using a comma-delimited list of server pool names.
- **SERVER_CATEGORY:** The name of a registered server category, used as part of server categorization. Oracle Clusterware Standard Clusters and Oracle Flex Clusters have default categories of hub and leaf. When you create a server pool, if you set a value for SERVER_CATEGORY, you cannot set a value for SERVER_NAMES. Only one of these parameters may have a nonempty value.
- **SERVER_NAMES:** Lists the candidate node names upon which servers reside that may be associated with a server pool. If this attribute is empty, Oracle Clusterware assumes that any server may be assigned to any server pool, to the extent allowed by other attributes, such as PARENT_POOLS.

Note: All attributes of the GENERIC server pool are read-only and cannot be modified. For the FREE server pool, only the IMPORTANCE and ACL attributes can be edited.

Server Pool Attribute Considerations

- You can use `srvctl` or `crsctl` to create server pools for databases and other applications.
 - If you use `crsctl` to create server pools, you can use the entire set of server pool attributes.
 - If you use `srvctl` to create a server pool, you can use only a subset of the server pool attributes:
 - `category`
 - `importance`
 - `min`
 - `max`
 - `serverpool`
 - `servers`
- Use `srvctl` to create server pools that host Oracle databases.
- Use `crsctl` to create server pools that host nondatabase resources such as middle tiers and applications.

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

You can use `srvctl` or `crsctl` to create server pools for databases and other applications, respectively. If you use `crsctl` to create server pools, you can use the entire set of server pool attributes. If you use `srvctl` to create a server pool, you can only use a subset of the server pool attributes. These server pool attributes are:

`-category`
`-importance`
`-min`
`-max`
`-serverpool`
`-servers`

The decision about which utility to use is based upon the type of resource being hosted in the server pool. You must use `crsctl` to create server pools that host nondatabase resources such as middle tiers and applications. You must use `srvctl` to create server pools that host Oracle databases. The `srvctl` utility prepends “ora.” to the name of the server pool.

GENERIC and FREE Server Pools

- When upgrading Clusterware, all nodes are placed in the **GENERIC** pool to ensure compatibility with earlier releases.
- The **GENERIC** server pool stores any server that is not in a top-level server pool and is not policy managed.
 - Servers hosting non-policy-managed applications are statically assigned to the **GENERIC** server pool.
- **FREE** server pool attributes are restricted, as follows:
 - **SERVER_NAMES**, **MIN_SIZE**, and **MAX_SIZE** cannot be edited by the user.
 - **IMPORTANCE** and **ACL** can be edited by the user.
- Configuration attributes of the **GENERIC** server pool cannot be edited.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

When Oracle Clusterware is installed, two server pools are created automatically: **GENERIC** and **FREE**. All servers in a new installation are assigned to the **FREE** server pool, initially. Servers move from **FREE** to newly defined server pools automatically. The **FREE** server pool contains servers that are not assigned to any other server pools. The attributes of the **FREE** server pool are restricted, as follows:

- **SERVER_NAMES**, **MIN_SIZE**, and **MAX_SIZE** cannot be edited by the user.
- **IMPORTANCE** and **ACL** can be edited by the user.

The **GENERIC** server pool stores any server that is not in a top-level server pool and is not policy managed. Servers that host non-policy-managed applications, such as administrator-managed databases, are statically assigned to the **GENERIC** server pool. The **GENERIC** server pool's attributes are restricted, as follows:

- Configuration attributes of the **GENERIC** server pool cannot be edited.

- You can only create administrator-managed databases in the Generic Pool, as long as the server you want to create the database on is one of the following:
 - Online and exists in the `GENERIC` server pool
 - Online and exists in the `FREE` server pool, in which case Oracle Clusterware moves the server into the `GENERIC` server pool
 - Online and exists in any other server pool and the user is either a cluster administrator or is allowed to use the server pool's servers, in which case, the server is moved into the `GENERIC` server pool
 - Offline and the user is a cluster administrator

Assignment of Servers to Server Pools

Assume that there are no servers in a cluster, all server pools are empty, and server pools are defined as follows:

| NAME | IMPORTANCE | MIN_SIZE | MAX_SIZE | PARENT_POOLS | EXCLUSIVE_POOLS |
|-------|------------|----------|----------|--------------|-----------------|
| sp1 | 1 | 1 | 10 | | |
| sp2 | 3 | 1 | 6 | | |
| sp3 | 2 | 1 | 2 | | |
| sp2_1 | 2 | 1 | 5 | sp2 | s123 |
| sp2_2 | 1 | 1 | 5 | sp2 | s123 |

1. Clusterware assigns host01 to sp2 because sp2 has the highest IMPORTANCE and its MIN_SIZE value has not yet been met.
2. Clusterware assigns host01 to sp2_1 but cannot assign host01 to sp2_2 because sp2_1 is configured to be exclusive with sp2_2.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Creating Server Pools with crsctl and srvctl

Use the `crsctl` utility or the `srvctl` utility to create additional server pools.

- Specifying attributes on the command line:

```
$ crsctl add serverpool SP1 -attr "MIN_SIZE=2, MAX_SIZE=5, IMPORTANCE=3"
```

```
$ srvctl add svrpool -serverpool SP1 -min 2 -max 5 -importance 3 -servers  
"server1,server2"
```

- Specifying attributes using a text file to supply them:

```
$ crsctl add serverpool SP1 -file /usr/local/bin/SP1_attributes.txt
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Use the `crsctl add serverpool` command or the `srvctl add svrpool` command to add a server pool to Oracle Clusterware. The only required attribute is `NAME`. The first example in the slide specifies the optional attributes on the command line using the `crsctl` utility. The attribute name and values must be enclosed in double quotation marks ("") and separated by commas. Do not use the `crsctl` utility for any server pools with names that begin with `ora` because these server pools are Oracle server pools. The `srvctl` utility is also shown. It allows only the `MIN_SIZE (-l)`, `MAX_SIZE (-u)`, `IMPORTANCE (-i)`, and `SERVER_NAMES (-n)` attributes to be specified. The second example in the slide specifies the attributes by identifying a text file that contains the attributes. Additional options to the `crsctl add serverpool` command include:

- `(-i)` : Causes the command to fail if the `crsd` process cannot complete the request immediately
- `(-f)` : Is the force option, which causes the `crsd` process to stop resources running on a server in another server pool and relocates that server into the server pool that you are adding

Note: New pools can be created only after the `GENERIC` pool has been deleted.

Managing Server Pools with `srvctl` and `crsctl`

Use the `crsctl` utility or the `srvctl` utility to delete and modify server pools.

- To delete server pools:

```
$ crsctl delete serverpool SP1
```

```
$ srvctl remove srvpool -serverpool SP1
```

- To modify server pools:

```
$ crsctl modify serverpool SP2 -attr "MIN_SIZE=4, MAX_SIZE=8, IMPORTANCE=7"
```

```
$ srvctl modify srvpool -serverpool SP2 -min 4 -max 8 -importance 7
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Moving Servers Between Server Pools

- Clusterware can move servers from other server pools into the server pool whose number of servers has fallen below the value for `MIN_SIZE`.
- Clusterware selects servers from other pools to move into the deficient pool that meet the following criteria:
 - For pools having a lower `IMPORTANCE` value than the deficient pool, servers can be moved from those pools even if it means that the number of servers falls below the value for the `MIN_SIZE` attribute.
 - For pools with equal or greater `IMPORTANCE`, Clusterware takes servers from those pools only if the number of servers in a pool is greater than the value of its `MIN_SIZE` attribute.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Managing Server Pools Using Default Attributes

- By default, each server pool is configured with the following attribute options for managing server pools:
 - `MIN_SIZE`: The minimum number of servers the server pool should contain
 - `MAX_SIZE`: The maximum number of servers the server pool should contain
 - `IMPORTANCE`: A number from 0 to 1000 (0 being least important) that ranks a server pool among all other server pools
- You can assign additional attributes to provide more granular management of server pools, as part of a *Cluster Configuration Policy*.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

By default, each server pool is configured with the following attribute options for managing server pools:

- `MIN_SIZE`: The minimum number of servers the server pool should contain.
If the number of servers in a server pool is below the value of this attribute, Oracle Clusterware automatically moves servers from elsewhere into the server pool until the number of servers reaches the attribute value.
- `MAX_SIZE`: The maximum number of servers the server pool should contain
- `IMPORTANCE`: A number from 0 to 1000 (0 being least important) that ranks a server pool among all other server pools in a cluster

In addition, you can assign additional attributes to provide more granular management of server pools, as part of a cluster configuration policy. Attributes such as `EXCLUSIVE_POOLS` and `SERVER_CATEGORY` can assist you to create policies for your server pools that enhance performance and build tuning design management into your server pool.

Server State Attributes

| Attribute | Purpose |
|---------------|---|
| NAME | The node name of the server |
| ACTIVE_POOLS | A space-delimited list of the names of the server pools to which a server belongs |
| STATE | The state of a server can be ONLINE, OFFLINE, LEAVING, JOINING, VISIBLE, and RECONFIGURING. |
| STATE_DETAILS | Additional details for STATE attributes |



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

- **LEAVING:** When a planned shutdown for a server begins, the state of the server transitions to LEAVING, making it unavailable for resource placement.
- **VISIBLE:** Servers that have Oracle Clusterware running, but not the Cluster Ready Services daemon (`crsd`), are put into the VISIBLE state. This usually indicates an intermittent issue or failure and Oracle Clusterware trying to recover (restart) the daemon. Oracle Clusterware cannot manage resources on servers while the servers are in this state.
- **RECONFIGURING:** When servers move between server pools due to server pool reconfiguration, a server is placed into this state if resources that ran on it in the current server pool must be stopped and relocated. This happens because resources running on the server may not be configured to run in the server pool to which the server is moving. As soon as the resources are successfully relocated, the server is put back into the ONLINE state.

STATE_DETAILS: This is a read-only attribute that Oracle Clusterware manages. The attribute provides additional details about the state of a server. Possible additional details about a server state are:

- **Server state:** ONLINE:
 - AUTOSTARTING RESOURCES: The resource autostart procedure (performed when a server reboots or the Oracle Clusterware stack is restarted) is in progress for the server.
 - AUTOSTART QUEUED: The server is waiting for the resource autostart to commence. When that happens, the attribute value changes to AUTOSTARTING RESOURCES.
- **Server state:** RECONFIGURING:
 - STOPPING RESOURCES: Resources that are restricted from running in a new server pool are stopping.
 - STARTING RESOURCES: Resources that can run in a new server pool are starting.
 - RECONFIG FAILED: One or more resources did not stop and thus the server cannot transition into the ONLINE state. At this point, manual intervention is required. You must stop or unregister resources that did not stop. After that, the server automatically transitions into the ONLINE state.
- **Server state:** JOINING:
 - CHECKING RESOURCES: Whenever a server reboots, the Oracle Clusterware stack restarts, or `crsd` on a server restarts, the policy engine must determine the current state of the resources on the server. While that procedure is in progress, this value is returned.

Server Categorization: Overview

- In previous releases, server pools were restricted to a set of basic attributes characterizing servers as belonging to a given pool.
- There was no way to distinguish between types of servers.
 - All servers were considered to be equal in relation to their processors, physical memory, and other characteristics.
- Server categorization enables you to organize servers into particular categories, using attributes such as processor types, memory, and other distinguishing system features.

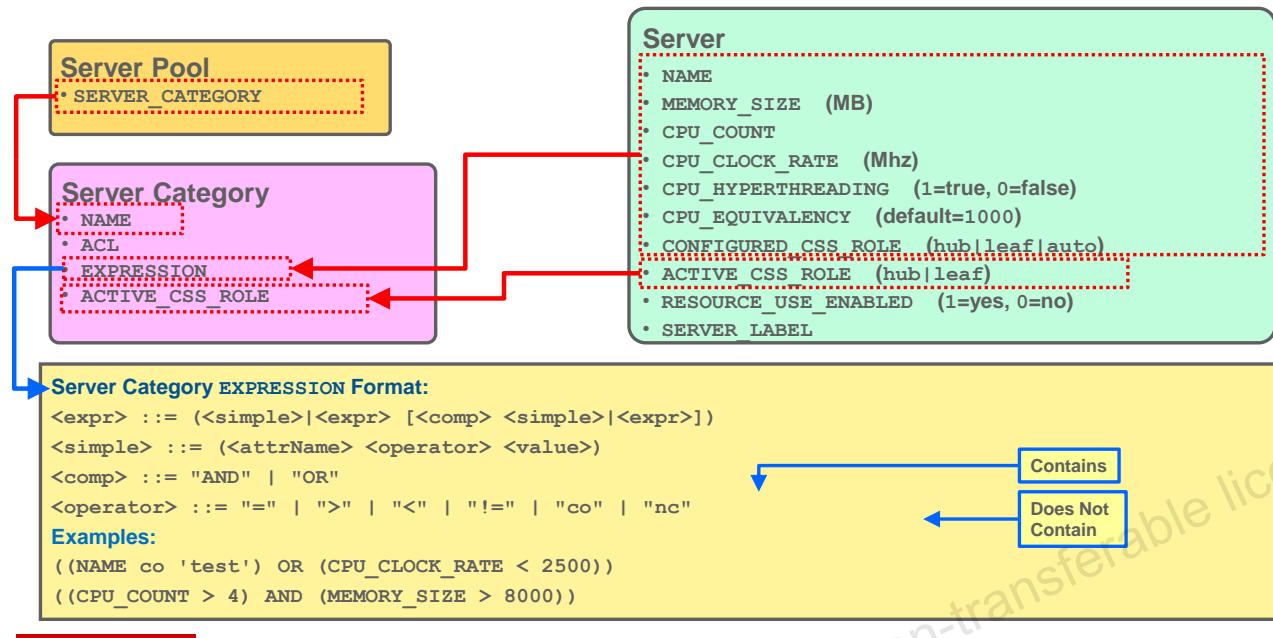


Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware 11g release 2 introduces server pools as a means for specifying resource placement and administering server allocation and access. Originally, server pools were restricted to a set of basic attributes characterizing servers as belonging to a given pool, with no way to distinguish between types of servers; all servers were considered to be equal in relation to their processors, physical memory, and other characteristics.

Server categorization enables you to organize servers into particular categories by using attributes such as processor types, memory, and other distinguishing system features. You can configure server pools to restrict eligible members of the pool to a category of servers, which share a particular set of attributes.

Server Categorization



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

In previous versions, the assignment of servers to server pools was based on the relative importance of each server pool and a few basic attributes, such as the minimum and maximum number of servers associated with the server pool. Because there was no way to differentiate between servers, all servers were assumed to be homogeneous with respect to CPU, memory, and other resources.

With Oracle Clusterware 12c, the notion of server categorization is introduced. Categorization allows servers to be differentiated and provides a mechanism for automatically controlling the composition of each sever pool. Server categorization works as follows:

- Every server contains a set of new attributes. Most of the attributes specify key physical characteristics of the server, such as `MEMORY_SIZE` and `CPU_COUNT`, or they contain configuration settings relating to Oracle Clusterware, such as `CONFIGURED_CSS_ROLE`. For a complete list of server attribute definitions, refer to the *Oracle Clusterware Administration and Deployment Guide 12c Release 1 (12.1)*.
- A new clusterware object defines server categories. The main attribute of each server category is an expression that is evaluated against the attributes of each server to determine whether the server belongs to the category.
- A new attribute, `SERVER_CATEGORY`, is added to the each server pool definition. This attribute allows a server category to be associated with each server pool, thereby governing which servers can be in the pool.

Administering Server Categorization: Server Attributes

- Most server attributes are automatically discovered by Oracle Clusterware.
- Example: Viewing attribute settings

```
$ crsctl status server host01 -f
NAME=host01
MEMORY_SIZE=4006
CPU_COUNT=1
CPU_CLOCK_RATE=2857
CPU_HYPERTHREADING=0
CPU_EQUIVALENCY=1000
DEPLOYMENT=other
CONFIGURED_CSS_ROLE=hub
RESOURCE_USE_ENABLED=1
SERVER_LABEL=UNAVAILABLE
PHYSICAL_HOSTNAME=UNAVAILABLE
STATE=ONLINE
ACTIVE_POOLS=Free
STATE_DETAILS=
ACTIVE_CSS_ROLE=hub
```

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Administering Server Categorization: Server Categories

- Creating a new server category:

```
$ crsctl add category <catName> -attr "<attrName>=<value>[,...]"
```

```
$ crsctl add category small -attr "EXPRESSION='(CPU_COUNT = 1)'"
```

- Modifying an existing server category:

```
$ crsctl modify category <catName> -attr "<attrName>=<value>[,...]"
```

```
$ crsctl modify category small -attr "ACTIVE_CSS_ROLE='hub'"
```

- Viewing a category:

```
$ crsctl status category <catName>
```

```
$ crsctl status category small  
NAME=small  
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r--  
ACTIVE_CSS_ROLE=hub  
EXPRESSION=(CPU_COUNT = 1)
```

- Deleting a category:

```
$ crsctl delete category <catName>
```

ORACLE

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The slide shows examples of the commands that can be used to create, modify, view, and delete a server category.

When creating or modifying a server category, note that the key attribute is the **EXPRESSION** that defines which servers can belong to the category. With the **ACTIVE_CSS_ROLE** attribute, administrators can specifically define different server categories for Hub Nodes and Leaf Nodes. The **ACTIVE_CSS_ROLE** attribute should not be referenced in the **EXPRESSION** string.

Administering Server Categorization: Server Categories

- Listing servers in a category:

```
$ crsctl status server -category <catName>
```

```
$ crsctl status server -category small  
NAME=host01  
STATE=ONLINE  
...
```

- Listing categories for a server:

```
$ crsctl status category -server <serverName>
```

```
$ crsctl status category -server host01  
NAME=ora.hub.category  
ACL=owner:root:rwx,pgrp:root:r-x,other::r--  
ACTIVE_CSS_ROLE=hub  
EXPRESSION=  
  
NAME=small  
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r--  
ACTIVE_CSS_ROLE=hub  
EXPRESSION=(CPU_COUNT = 1)
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Administering Server Categorization: Server Pools

- Specifying the SERVER_CATEGORY attribute:

```
$ crsctl add serverpool hr -attr "SERVER_CATEGORY='medium'" ...
```

```
$ crsctl modify serverpool dev -attr "SERVER_CATEGORY='small'"
```

- Viewing the SERVER_CATEGORY attribute:

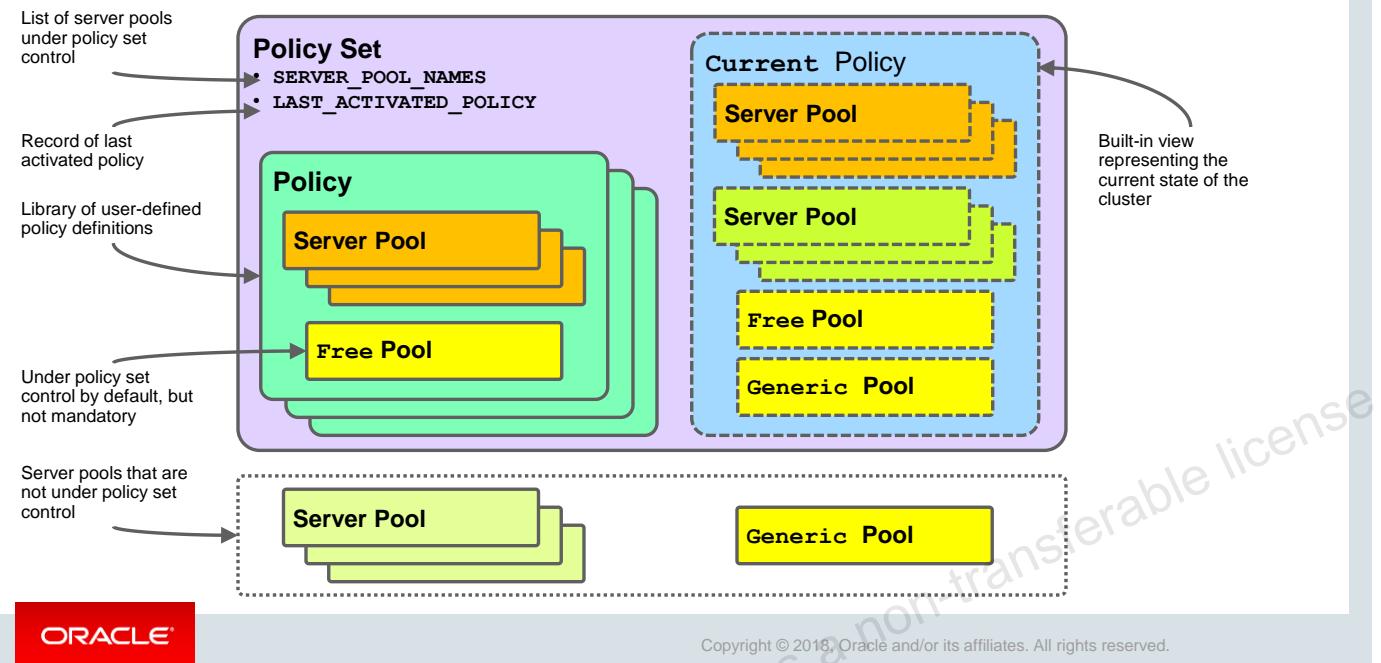
```
$ crsctl status serverpool dev -f
NAME=dev
IMPORTANCE=0
MIN_SIZE=0
MAX_SIZE=-1
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r--
SERVER_CATEGORY=small
ACTIVE_SERVERS=host01 host02
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Server categories are applied to server pools by using the new SERVER_CATEGORY attribute. This attribute can be specified for new and existing server pools, as shown in the examples in the slide. To view the setting of the SERVER_CATEGORY attribute, use the crsctl status serverpool command with the -f option.

Policy Set: Overview



The policy set always contains a special built-in policy, named `Current`, representing the configuration that is currently in effect. The `Current` policy includes all server pools that are not under policy set control. When a user-defined policy is activated, its attributes are reflected in the `Current` policy. Over time, the `Current` policy may cease to reflect the last activated policy because of changes made outside the policy set; for example, when a server pool associated with a release 11.2 policy-managed database is added to the system.

In previous versions, two built-in server pools existed: `Generic` and `Free`. With Oracle Clusterware 12c, these built-in server pools remain; however, they are handled differently. By default, the `Free` server pool is implicitly controlled by the policy set. However, administrators can choose to remove the `Free` server pool from the `SERVER_POOL_NAMES` list if they want to place the `Free` pool outside direct policy set control. The `Generic` server pool is never under direct policy set control. It is listed as a server pool in the `Current` policy view.

Policy-Based Cluster Management and QoS Management

Two methods for configuring and running policy-based cluster management:

- User-defined policy management
 - Clusterware administrators manually configure the policy set.
 - Clusterware administrators activate different policies as required.
 - Can use a job scheduling system to automatically activate specific policies at different times
- Quality of Service (QoS) Management
 - QoS Management interfaces are used to configure the policy set.
 - Administrators cannot directly modify the policy set.
 - QoS Management automatically adjusts resource allocations in response to workload demands.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

There are two distinct modes of operation that apply to policy-based cluster management.

With user-defined policy management, clusterware administrators manually configure the policy set, policies, server pools, and their attributes. After configuration, it is the responsibility of the clusterware administrator to activate the required policy. Policies can also be activated automatically by using a job scheduling system or another program that uses the supplied commands and application programming interfaces (APIs).

During implementation of QoS Management, the QoS Management interfaces are used to configure a QoS Management policy set, which also contains a clusterware policy set definition. When QoS Management is activated, the associated clusterware policy set definition is activated and locked so that clusterware administrators cannot manually modify the policy set. This allows QoS Management to automatically adjust policy settings to fulfill the prescribed performance objectives.

In essence, QoS Management extends the functionality that is available with user-based policy management. Direct policy manipulation outside the QoS Management interfaces is not possible while QoS Management is enabled.

The remainder of this lesson focuses on user-based policy management as a new feature of Oracle Clusterware 12c. QoS Management is outside the scope of this course.

Viewing the Policy Set

```
$ crsctl status policyset
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r-x
LAST_ACTIVATED_POLICY=
SERVER_POOL_NAMES=Free
POLICY
  NAME=Current
  DESCRIPTION=This policy is built-in and managed automatically to reflect current
configuration
  SERVERPOOL
    NAME=Free
    ACTIVE_SERVERS=host01 host02
    EXCLUSIVE_POOLS=
    IMPORTANCE=0
    MAX_SIZE=-1
    MIN_SIZE=0
    PARENT_POOLS=
    SERVER_CATEGORY=
    SERVER_NAMES=
...
...
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

A policy set is implicitly created in every cluster. Because there is exactly one policy set per cluster, policy sets cannot be created or deleted and no name is required to identify the policy set.

The `crsctl status policyset` command can be used to view the current policy set attributes, including all policies and server pools that are defined on the cluster. The output also includes information about the current state of the cluster, which is listed under the special built-in policy named `Current`.

Configuring a User-Defined Policy Set: Method 1

1. Set the SERVER_POOL_NAMES policy set attribute:

```
$ crsctl modify policyset -attr "SERVER_POOL_NAMES='dev test'"
```

2. Add the policies:

```
$ crsctl add policy day -attr "DESCRIPTION='The day policy'"  
$ crsctl add policy night -attr "DESCRIPTION='The night policy'"
```

3. Set the server pool attributes for each policy:

```
$ crsctl modify serverpool dev -attr  
"IMPORTANCE=10,MAX_SIZE=2,MIN_SIZE=1,SERVER_CATEGORY=small" -policy day  
  
$ crsctl modify serverpool test -attr "IMPORTANCE=5,MAX_SIZE=2,MIN_SIZE=1" -policy day  
  
$ crsctl modify serverpool dev -attr  
"IMPORTANCE=5,MAX_SIZE=2,MIN_SIZE=0,SERVER_CATEGORY=small" -policy night  
  
$ crsctl modify serverpool test -attr "IMPORTANCE=10,MAX_SIZE=2,MIN_SIZE=2" -policy night
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The policy set can be configured by using the `crsctl` command-line utility. The slide outlines one method and shows a series of example commands. The procedure is:

1. Set the SERVER_POOL_NAMES policy set attribute. This attribute formally defines the scope of the server pools that are controlled by the policy set. In addition, any server pool named in the SERVER_POOL_NAMES policy set attribute is implicitly created if it did not previously exist.
2. Add the policies. Each policy that is created in this phase is automatically created with a default set of attributes describing each of the server pools named in the previous step.
3. Set the attributes for the server pools in each policy.

Configuring a User-Defined Policy Set: Method 2

1. Create a policy set definition file:

```
$ cat policyset.txt
SERVER_POOL_NAMES=dev test
POLICY
  NAME=day
  DESCRIPTION=The day policy
  SERVERPOOL
    NAME=dev
    IMPORTANCE=10
    MAX_SIZE=2
    MIN_SIZE=1
    SERVER_CATEGORY=small
  SERVERPOOL
    NAME=test
    IMPORTANCE=5
    MAX_SIZE=2
    MIN_SIZE=1
```

```
POLICY
  NAME=night
  DESCRIPTION=The night policy
  SERVERPOOL
    NAME=dev
    IMPORTANCE=5
    MAX_SIZE=2
    MIN_SIZE=0
    SERVER_CATEGORY=small
  SERVERPOOL
    NAME=test
    IMPORTANCE=10
    MAX_SIZE=2
    MIN_SIZE=2
```

2. Modify the policy set:

```
$ crsctl modify policyset -file policyset.txt
```

ORACLE

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Modifying a User-Defined Policy Set

- Method 1
 - Modify the policy set directly by using `crsctl` commands.
 - Examples:

```
$ crsctl add policy day -attr "DESCRIPTION='The day policy'"
```

```
$ crsctl modify serverpool dev
-attr "IMPORTANCE=10,MAX_SIZE=2,MIN_SIZE=1,SERVER_CATEGORY=small"
-policy day
```

- Method 2

1. Create a policy set definition file:

```
$ crsctl status policyset -file policyset.txt
```

2. Edit the policy set definition file.

3. Modify the policy set:

```
$ crsctl modify policyset -file policyset.txt
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

There are essentially two ways to modify a user-defined policy set:

1. Administrators can directly modify the policy set by using specific `crsctl` commands. Examples of direct manipulation are listed in the slide.
By default, attempts to modify policies and server pools fail if the modification causes a cluster-managed resource, such as a database instance, to shut down. Alternatively, the `-f` command line option can be added to force the change. Also, what-if command evaluation (described later) can be used to test the effect of a change prior to implementation.
2. Administrators can use the `crsctl status policyset` command with the `-file` option to dump the current policy set definition into a text file. The resulting text file can then be modified to define an updated policy set. Finally, the updated policy set can be loaded into the system by using the `crsctl modify policyset` command with the `-file` option.

This method also provides an effective way to configure the policy set when administrators start with existing server pool definitions that were created with Oracle Clusterware, release 11.2.

Activating a User-Defined Policy

- Set the LAST ACTIVATED POLICY policy set attribute:

```
$ crsctl modify policyset -attr "LAST_ACTIVATED_POLICY='day'"
```

- Verify the policy settings:

```
$ crsctl status policyset
...
LAST_ACTIVATED_POLICY=day
SERVER_POOL_NAMES=dev test Free
POLICY
  NAME=Current
...
  SERVERPOOL
    NAME=dev
    ACTIVE_SERVERS=host01
...
  SERVERPOOL
    NAME=test
    ACTIVE_SERVERS=host02
...
...
```

```
$ crsctl status policy -active
POLICY
  NAME=Current
...
  SERVERPOOL
    NAME=dev
    ACTIVE_SERVERS=host01
...
  SERVERPOOL
    NAME=test
    ACTIVE_SERVERS=host02
...
```

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

After the policy set is initially configured, none of the defined policies are active. To activate a policy, the LAST_ACTIVATED_POLICY policy set attribute must be set. An example is shown at the top of the slide. The active policy can be changed at will by using this command. Alternatively, system administrators can use a job scheduling system or other management tools to automatically activate different policies based on different times of day or other circumstances.

When a new policy is activated, nodes are automatically reassigned to server pools, and relevant resources are automatically started or stopped in line with the new active policy.

You can examine the active policy by using the crsctl status commands shown in the slide. Examine the LAST_ACTIVATED_POLICY policy set attribute, and also check the server assignments in each server pool along with other server pool attributes to verify the policy settings.

Load-Aware Resource Placement

- You can configure a database so that Clusterware is aware of the CPU requirements and limits for the database.
- Clusterware uses this to place the database on servers that have a sufficient number of CPUs, enough memory, or both.
- For database resources, you can configure the CPU or memory values by using the `CPU_COUNT` and `MEMORY_TARGET` parameters.

```
$ srvctl modify database -db db_unique_name -cpucount cpu_count \  
-memorytarget memory_target
```

- Configuration Requirements:
 - For CPU, the Resource Manager must be enabled for Instance Caging.
 - For memory, Automatic Memory Management must be used for the database.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

You can configure a database so that Oracle Clusterware is aware of the CPU requirements and limits for the given database. Clusterware uses this information to place the database resource only on servers that have sufficient number of CPUs, amount of memory, or both.

If you have configured resources with CPU or memory requirements in Clusterware, it will attempt to start those resources only on servers that meet those requirements. For database resources, specifically, you can configure the CPU or memory values into the `CPU_COUNT` and `MEMORY_TARGET` instance parameters, respectively.

When you add or modify a database instance, you can configure load-aware resource placement, as shown in the slide above.

In the example, `cpu_count` refers to the number of workload CPUs and `memory_target` refers to the target memory, in MB, used by the resource.

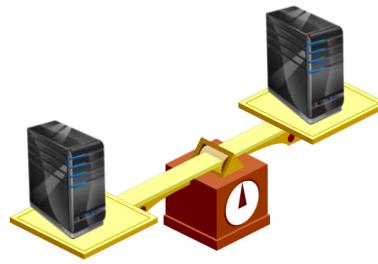
If Resource Manager is enabled in the database, Oracle Clusterware sets the `CPU_COUNT` parameter to the value of `-cpucount`. Also, if Automatic Memory Management is enabled for the database, Oracle Clusterware sets the `MEMORY_TARGET` database parameter to the value of `-memorytarget`.

Note: Configuring the instance parameters requires the following:

- For CPU (Instance Caging), the Resource Manager must be enabled
- For memory, Automatic Memory Management must be used for the database

Server Weight-Based Node Eviction

- Clusterware can be configured to select which nodes to terminate or evict in the event of an interconnect failure.
- In a split-brain situation, Clusterware applies certain rules to select the surviving nodes, potentially evicting a node.
- You can affect the outcome of these decisions by adding a value to a database instance or cluster node.
- The server weight-based node eviction mechanism helps to identify the node or the group of nodes to be evicted based on additional information about the load on those servers.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

You can configure the Clusterware failure recovery mechanism to choose which cluster nodes to terminate or evict in the event of a private network (cluster interconnect) failure. In a split-brain situation, where a cluster experiences a network split, partitioning the cluster into disjoint cohorts, Oracle Clusterware applies certain rules to select the surviving cohort, potentially evicting a node that is running a critical, singleton resource.

You can affect the outcome of these decisions by adding value to a database instance or node; thus, when Clusterware must decide whether to evict or terminate, it will consider these factors and attempt to ensure that all critical components remain available. You can configure weighting functions to add weight to critical components in your cluster, giving Clusterware added input when it decides which nodes to evict when resolving a split-brain situation.

You may want to ensure that specific nodes survive the tie-breaking process because of certain hardware characteristics, or that certain resources survive, because of particular databases or services. You can assign weight only to administrator-managed nodes or to servers or applications that are registered Clusterware resources.

Weight contributes to importance of a component and influences the choice that Oracle Clusterware makes when managing a split-brain situation. With other critical factors being equal between the various cohorts, Oracle Clusterware chooses the heaviest cohort to survive.

Assigning Weight to Servers and Resources

- To assign weight to a server by using `crsctl`, use the `css_critical yes` parameter.

```
$ crsctl set server css_critical yes
CRS-4416: Server attribute 'CSS_CRITICAL' successfully changed.
Restart Oracle High Availability Services for new value to take effect.
```

- To assign weight to database instances or services with `srvctl`, you use the `--css_critical yes` parameter.

```
$ srvctl add database -db orcl ... --css_critical yes
$ srvctl modify service ... --css_critical yes
```

- To assign weight to non `ora.*` resources, use the `-attr "CSS_CRITICAL=yes"` parameter.

```
$ crsctl add resource -attr "CSS_CRITICAL=yes"
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

You can assign weight to various components as follows:

- To assign weight to database instances or services, you use the `--css_critical yes` parameter with the `srvctl add database` or `srvctl add service` commands when adding a database instance or service. You can also use the parameter with the `srvctl modify database` and `srvctl modify service` commands. You can use this parameter only in administrator managed node. If the node becomes policy managed at some point, this parameter will no longer apply.
- To assign weight to non `ora.*` resources, use the `-attr "CSS_CRITICAL=yes"` parameter with the `crsctl add resource` and `crsctl modify resource` commands when you are adding or modifying resources.
- To assign weight to a server, use the `css_critical yes` parameter with the `crsctl set server` command.

You must restart the Clusterware stack on the node for the values to take effect. This does not apply to resources where the changes take effect without having to restart the resource.

If you change the environment from administrator managed to policy managed, or a mix of the two, any weight that you have assigned is stored, but is not considered, meaning that it will no longer apply or be considered unless and until you reconfigure the cluster back to being administrator managed.



Quiz

Which statements about server categorization are correct?

- a. Server categorization provides a mechanism to control which servers can be in a server pool.
- b. A server category must contain one or more servers.
- c. A server can belong to only one server category at a time.
- d. Servers can be categorized with user-defined expressions that combine various server attributes.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.



Quiz

Which statements about policy-based cluster management are correct?

- a. Administrators can create multiple policy sets for different workloads and priorities.
- b. Administrators can create multiple policies for different workloads and priorities.
- c. The policy set automatically manages all server pools defined in the cluster.
- d. The policy set automatically manages the server pools identified in the SERVER_POOL_NAMES attribute, and policy changes have no effect on other server pools.
- e. The policy set automatically manages the server pools identified in the SERVER_POOL_NAMES attribute, and policy changes may indirectly affect other server pools.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.



Quiz

The Current policy _____ matches the policy definition specified in the LAST_ACTIVATED_POLICY policy set attribute.

- a. Sometimes
- b. Always
- c. Never



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Answer: c

The Current policy never matches the policy definition specified in the LAST_ACTIVATED_POLICY policy set attribute because it includes the Generic built-in server pool, which is never under policy set control. Also, the Current policy contains the current state of server pools not under policy set control. It is fair to say that the attributes of the LAST_ACTIVATED_POLICY policy may be reflected in the Current policy; however, those attributes may also vary over time because of changes made outside the policy set.

Summary

In this lesson, you should have learned how to:

- Describe the architecture and components of policy-based cluster management
- Administer server categorization
- Administer a policy set
- Activate a policy
- Implement load-ware resource placement
- Implement server weight-based node eviction



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Practice 8 Overview: Using Policy-Based Cluster Management

This practice covers the following topics:

- Configuring server categories and the policy set
- Examining the effect of various changes to verify the dynamic nature of policy-based cluster management
- Examining how easy it is to activate policies



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Unauthorized reproduction or distribution prohibited. Copyright© 2019, Oracle and/or its affiliates.

GANG LIU (gangl@baylorhealth.edu) has a non-transferable license
to use this Student Guide.

Upgrading and Patching Grid Infrastructure

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe the types of patches and upgrades available
- Plan for rolling patches and rolling upgrades
- Compare software versions with the active version
- Install a patch with the `opatch` utility



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Clusterware Upgrading and Patching: Overview

- In-place patching replaces the Oracle Clusterware software with the newer version in the same Grid home.
- An out-of-place upgrade has both software versions present in different Grid homes, but only one version is active.
- Only out-of-place upgrades are supported because Oracle Clusterware 12c must have its own, new Grid home.
- For Oracle Clusterware 12c, Oracle supports in-place or out-of-place patching.
- Patch bundles and one-off patches are supported for in-place patching.
 - Only patch sets and major point releases for out-of-place upgrades are supported.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

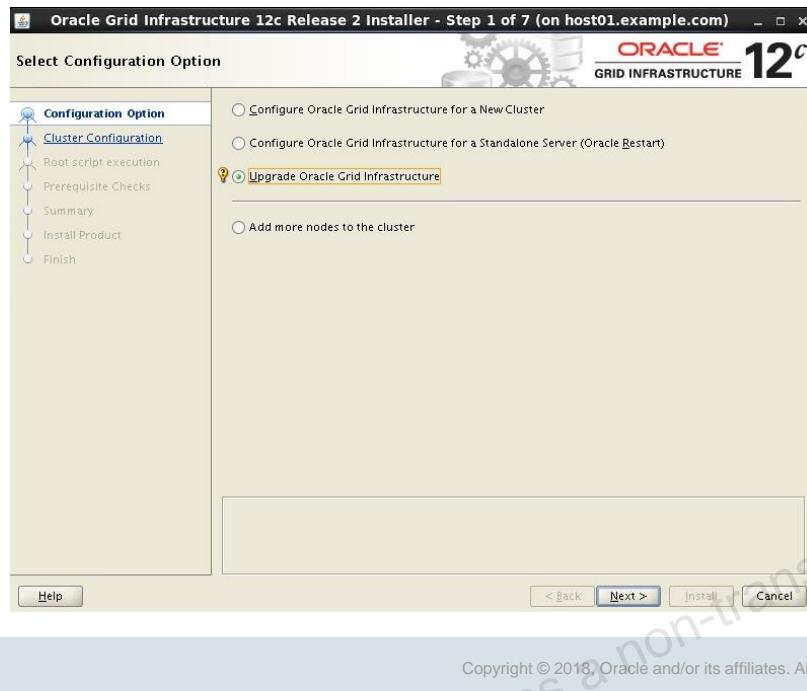
Oracle supports out-of-place upgrades, only, because Oracle Clusterware 12c must have its own, new Grid home. For Oracle Clusterware 12c, Oracle supports in-place or out-of-place patching. Oracle supports patch bundles and one-off patches for in-place patching but only supports patch sets and major point releases for out-of-place upgrades.

In-place patching replaces the Oracle Clusterware software with the newer version in the same Grid home. Out-of-place upgrade has both versions of the same software present on the nodes at the same time, in different Grid homes, but only one version is active.

Rolling upgrades avoid down time and ensure continuous availability of Oracle Clusterware while the software is upgraded to the new version. When you upgrade to Oracle Clusterware 12c, Oracle Clusterware and Oracle ASM binaries are installed as a single Oracle Home called Oracle Grid Infrastructure. You can upgrade Oracle Clusterware in a rolling manner from Oracle Clusterware 10g and Oracle Clusterware 11g; however, you can only upgrade Oracle ASM in a rolling manner from Oracle Database 11g release 1.

Oracle supports force upgrades in cases where some nodes of the cluster are down.

Oracle Grid Infrastructure Upgrade



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

With an out-of-place upgrade, the installer installs the newer version in a separate Oracle Clusterware home. Both versions of Oracle Clusterware are on each cluster member node, but only one version is active.

Rolling upgrade avoids downtime and ensure continuous availability while the software is upgraded to a new version.

If you have separate Oracle Clusterware homes on each node, then you can perform an out-of-place upgrade on all nodes, or perform an out-of-place rolling upgrade, so that some nodes are running Oracle Clusterware from the earlier version Oracle Clusterware home, and other nodes are running Oracle Clusterware from the new Oracle Clusterware home.

An in-place upgrade of Oracle Grid Infrastructure is not supported.

Options for Oracle Grid Infrastructure Upgrades

- Supported upgrade paths for Oracle Grid Infrastructure for this release are:
 - Oracle Grid Infrastructure upgrade from releases 11.2.0.3 and 11.2.0.4 to Oracle Grid Infrastructure 12cRelease 2 (12.2).
 - Oracle Grid Infrastructure upgrade from Oracle Grid Infrastructure 12c Release 1 (12.1) to Oracle Grid Infrastructure 12c Release 2 (12.2).
- Upgrade options from Oracle Grid Infrastructure 11g and Oracle Grid Infrastructure 12c Release 1 (12.1) to Oracle Grid Infrastructure 12c Release 2 (12.2) include the following:
 - Oracle Grid Infrastructure rolling upgrade which involves upgrading individual nodes without stopping Oracle Grid Infrastructure on other nodes in the cluster
 - Oracle Grid Infrastructure non-rolling upgrade by bringing the cluster down and upgrading the complete cluster



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Understand the upgrade options for Oracle Grid Infrastructure in this release. When you upgrade to Oracle Grid Infrastructure 12c Release 2 (12.2), you upgrade to an Oracle Flex Cluster configuration.

Supported upgrade paths for Oracle Grid Infrastructure for this release are:

- Oracle Grid Infrastructure upgrade from releases 11.2.0.3 and 11.2.0.4 to Oracle Grid Infrastructure 12cRelease 2 (12.2).
- Oracle Grid Infrastructure upgrade from Oracle Grid Infrastructure 12c Release 1 (12.1) to Oracle Grid Infrastructure 12c Release 2 (12.2).

Upgrade options from Oracle Grid Infrastructure 11g and Oracle Grid Infrastructure 12c Release 1 (12.1) to Oracle Grid Infrastructure 12c Release 2 (12.2) include the following:

- Oracle Grid Infrastructure rolling upgrade which involves upgrading individual nodes without stopping Oracle Grid Infrastructure on other nodes in the cluster
- Oracle Grid Infrastructure non-rolling upgrade by bringing the cluster down and upgrading the complete cluster

Note: When you upgrade to Oracle Grid Infrastructure 12c Release 2 (12.2), you upgrade to an Oracle Standalone Cluster configuration. If storage for OCR and voting files is other than Oracle ASM, you need to migrate OCR and voting files to Oracle ASM before upgrading to Oracle Grid Infrastructure 12c Release 2 (12.2).

Pre-Upgrade Tasks

- For each node, use Cluster Verification Utility to ensure that you have completed preinstallation steps.
- Ensure that you have the following information:
 - An Oracle base location for Oracle Clusterware
 - A Grid home location different from your existing location
 - SCAN name and addresses, and other network addresses
 - Privileged O/S users and groups as described in the lesson titled “Grid Infrastructure Preinstallation Tasks”
 - `root` user access, to run scripts as root during installation
- Unset Oracle variables in the installation owner’s environment:

```
$ unset ORACLE_BASE
$ unset ORACLE_HOME
$ unset ORACLE_SID
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Complete the following tasks before starting an upgrade:

1. For each node, use Cluster Verification Utility to ensure that you have completed preinstallation steps. It can generate `fixup` scripts to help you to prepare servers. In addition, the installer will help you to ensure all required prerequisites are met. Ensure that you have information you will need during installation, including the following:
 - An Oracle base location for Oracle Clusterware
 - An Oracle Grid Infrastructure home location that is different from your existing Oracle Clusterware location
 - SCAN name and addresses, and other network
 - Privileged operating system users and groups as described in the lesson titled “Grid Infrastructure Preinstallation Tasks”
 - Root user access, to run scripts as root during installation
2. For the installation owner running the installation, if you have environment variables set for the existing installation, then unset the environment variables `$ORACLE_HOME` and `$ORACLE_SID`, because these environment variables are used during upgrade.

Example:

```
$ unset ORACLE_BASE
$ unset ORACLE_HOME
$ unset ORACLE_SID
```

Moving Oracle Clusterware Files to Oracle ASM

1. As Oracle Grid Infrastructure owner (`grid`), create the Oracle ASM disk group using ASMCA.
2. As `grid` user, move the voting files to the Oracle ASM disk group you created.

```
$ crsctl replace votedisk +DATA
```

3. As `grid` user, check the Oracle Cluster Registry (OCR) status.

```
$ ocrcheck
```

4. As `root` user, move the OCR files to the Oracle ASM disk group you created.

```
# ocrconfig -add +DATA
```

5. As `root` user, delete the Oracle Clusterware files from the NFS location.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

If Oracle Cluster Registry (OCR) and voting files are stored on Network File System (NFS), then move these files to Oracle ASM disk groups before upgrading Oracle Grid Infrastructure.

1. As Oracle Grid Infrastructure installation owner (`grid`), create the Oracle ASM disk group using ASMCA. Follow the steps in the ASMCA wizard to create the Oracle ASM disk group, for example, DATA.
2. As `grid` user, move the voting files to the Oracle ASM disk group you created.
`$ crsctl replace votedisk +DATA`
3. As `grid` user, check the Oracle Cluster Registry (OCR) status.
`$ ocrcheck`
4. As `root` user, move the OCR files to the Oracle ASM disk group you created.
`# ocrconfig -add +DATA`
5. As `root` user, delete the Oracle Clusterware files from the NFS location.
`# ocrconfig -delete <ocr_file_path_previously_on_nfs>`

Using CVU to Validate Readiness for Clusterware Upgrades

- You can use Cluster Verification Utility (CVU) to assist you with system checks in preparation for starting an upgrade.
- You can run upgrade validations in one of two ways:
 - Run OUI, and allow the CVU validation built into OUI to perform system checks and generate fixup scripts.
 - Run the CVU manual script `runcluvfy.sh` to perform system checks and generate fixup scripts.
- Assuming a two-node cluster, verify the upgrade readiness of the current Clusterware installation, running a command similar to the following one from the upgrade location:

```
$ ./runcluvfy.sh stage -pre crsinst -upgrade -rolling -n host01,host02 -  
rolling -src_crshome /u01/app/11.2.0/grid -dest_crshome /u01/app/12.2.0/grid  
-dest_version 12.2.0.1 -fixup -verbose
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

You can use Cluster Verification Utility (CVU) to assist you with system checks in preparation for starting an upgrade. CVU runs the appropriate system checks automatically, and either prompts you to fix problems, or provides a fixup script to be run on all nodes in the cluster before proceeding with the upgrade. You can run upgrade validations in one of two ways:

- Run OUI, and allow the CVU validation built into OUI to perform system checks and generate fixup scripts.
- Run the CVU script `cluvfy.sh` to perform system checks and generate fixup scripts.

To use OUI to perform preinstall checks and generate fixup scripts, run the installation as you normally would. To use the `cluvfy.sh` command-line script for CVU, navigate to the staging area for the upgrade, where the `runcluvfy.sh` command is located, and run the `runcluvfy.sh stage -pre crsinst -upgrade` command to check the readiness of your Oracle Clusterware installation for upgrades. Running `runcluvfy.sh` with the `-pre crsinst -upgrade` flags performs system checks to confirm if the cluster is in a correct state for upgrading from an existing Clusterware installation. You can verify that the permissions required for installing Oracle Clusterware have been configured on the `host01` and `host02` nodes by running a command similar to the following:

```
$ ./runcluvfy.sh stage -pre crsinst -upgrade -n node1,node2 -rolling -  
src_crshome /u01/app/11.2.0/grid -dest_crshome /u01/app/12.2.0/grid -  
dest_version 12.2.0.1 -fixup -verbose
```

Understanding Rolling Upgrades Using Batches

- Upgrades from earlier releases require that you upgrade the entire cluster.
- You cannot select or deselect individual nodes to upgrade.
- Oracle recommends that you leave RAC instances running.
 - When the `root` script is run on each node, that node's instances are shut down and started up again by the `rootupgrade.sh` script.
- You can use the `root` user automation to automate running the `rootupgrade.sh` script during the upgrade.
- When you use `root` automation, you can divide the nodes into batches and then start upgrades of these batches.
- Between batches, services can be moved from un-upgraded to upgraded nodes so that services are not affected by the upgrade.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Upgrades from earlier releases require that you upgrade the entire cluster. You cannot select or deselect individual nodes for upgrade. Oracle does not support attempting to add additional nodes to a cluster during a rolling upgrade.

Oracle recommends that you leave Oracle RAC instances running. When you start the `root` script on each node, that node's instances are shut down and then started up again by the `rootupgrade.sh` script. If you upgrade from Oracle Grid Infrastructure 11g Release 11.2.0.1 and later to any later release of Oracle Grid Infrastructure, all nodes are selected by default.

You can use `root` user automation to automate running the `rootupgrade.sh` script during the upgrade. When you use `root` automation, you can divide the nodes into groups, or batches, and start upgrades of these batches. Between batches, you can move services from nodes running the previous release to the upgraded nodes, so that services are not affected by the upgrade. Oracle recommends that you use `root` automation, and allow the `rootupgrade.sh` script to stop and start instances automatically. You can also continue to run `root` scripts manually.

The following restrictions apply when selecting nodes in batches for upgrade:

- You can pool nodes in batches for upgrade, up to a maximum of three batches.
- The local node, where Oracle Universal Installer (OUI) is running, must be upgraded in batch one.
- Hub and Leaf Nodes cannot be upgraded in the same batch.
- All Hub Nodes must be upgraded before starting the upgrade of Leaf Nodes.

Performing a Rolling Upgrade from an Earlier Release

1. As grid user, download the Oracle Grid Infrastructure image files and extract the files to the Grid home.

```
$ mkdir -p /u01/app/12.2.0/grid  
$ chown grid:oinstall /u01/app/12.2.0/grid  
$ cd /u01/app/12.2.0/grid  
$ unzip -q <download_location>/grid_home.zip
```

2. Start the Oracle Grid Infrastructure wizard by running the following command.

```
$ <Grid_home>/gridSetup.sh
```

3. Select the “Upgrade Oracle Grid Infrastructure.” option.
4. On the Node Selection page, select all nodes.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Use the following procedure to upgrade the cluster from an earlier release. At any time during the upgrade, if you have a question about what you are being asked to do, or what input you are required to provide during upgrade, click the Help button on the installer page.

You should have your network information, storage information, and operating system users and groups available to you before you start upgrade, and you should be prepared to run root scripts.

1. As grid user, download the Oracle Grid Infrastructure image files and extract the files to the Grid home.

```
$ mkdir -p /u01/app/12.2.0/grid  
$ chown grid:oinstall /u01/app/12.2.0/grid  
$ cd /u01/app/12.2.0/grid  
$ unzip -q <download_location>/grid_home.zip
```

2. Start the Oracle Grid Infrastructure wizard by running the following command.

```
$ <Grid_home>/gridSetup.sh
```

3. Select the following configuration option.

- **Upgrade Oracle Grid Infrastructure:** Select this option to upgrade Oracle Grid Infrastructure (Oracle Clusterware and Oracle ASM).

4. On the node selection page, select all nodes.

Performing a Rolling Upgrade from an Earlier Release

5. Select installation options as prompted. Oracle recommends that you configure `root` script automation.
6. Run `root` scripts, using either automatically or manually:
 - Running `root` scripts automatically
 - Running `root` scripts manually
7. Update any scripts or applications that use utilities, libraries, or other files that reside in the Oracle Clusterware and Oracle ASM homes.
8. Update the Oracle Enterprise Manager target parameters as described in the topic *Updating Oracle Enterprise Manager Cloud Control Target Parameters*.

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

5. Select installation options as prompted. Oracle recommends that you configure root script automation, so that the `rootupgrade.sh` script can be run automatically during the upgrade.
6. Run root scripts, using either automatically or manually:
 - Running root scripts automatically: If you have configured root script automation, then use the pause between batches to relocate services from the nodes running the previous release to the new release.
 - Running root scripts manually: If you have not configured root script automation, then when prompted, run the `rootupgrade.sh` script on each node in the cluster that you want to upgrade.

If you run root scripts manually, then run the script on the local node first. The script shuts down the earlier release installation, replaces it with the new Oracle Clusterware release, and starts the new Oracle Clusterware installation. After the script completes successfully, you can run the script in parallel on all nodes except for one, which you select as the last node. When the script is run successfully on all the nodes except the last node, run the script on the last node. When upgrading from 12.1 Oracle Flex Cluster, Oracle recommends that you run the `rootupgrade.sh` script on all Hub Nodes before running it on Leaf Nodes.

- Unauthorized reproduction or distribution prohibited. Copyright© 2019, Oracle and/or its affiliates
7. Because the Oracle Grid Infrastructure home is in a different location than the former Oracle Clusterware and Oracle ASM homes, update any scripts or applications that use utilities, libraries, or other files that reside in the Oracle Clusterware and Oracle ASM homes.
 8. Update the Oracle Enterprise Manager target parameters as described in the topic *Updating Oracle Enterprise Manager Cloud Control Target Parameters*.

Note:

- At the end of the upgrade, if you set the Oracle Cluster Registry (OCR) backup location manually to the earlier release Oracle Clusterware home (CRS home), then you must change the OCR backup location to the new Oracle Grid Infrastructure home (Grid home). If you did not set the OCR backup location manually, then the backup location is changed for you during the upgrade.
- Because upgrades of Oracle Clusterware are out-of-place upgrades, the previous release Oracle Clusterware home cannot be the location of the current release OCR backups. Backups in the old Oracle Clusterware home can be deleted.
- If the cluster being upgraded has a single disk group that stores the OCR, OCR backup, Oracle ASM password, Oracle ASM password file backup, and the Grid Infrastructure Management Repository (GIMR), then Oracle recommends that you create a separate disk group or use another existing disk group and store the OCR backup, the GIMR and Oracle ASM password file backup in that disk group.

Completing a Clusterware Upgrade When Nodes Become Unreachable

- If some nodes become unreachable in the middle of an upgrade, you cannot complete the upgrade.
- This is because the `rootupgrade.sh` script will not have run on the unreachable nodes.
- Confirm that the upgrade is incomplete by running:

```
$ crsctl query crs activeversion  
Oracle Clusterware active version on the cluster is [11.2.0.3.0]
```

- Resolve this problem by running the `rootupgrade.sh` command with the `-force` flag:

```
# /u01/app/12.2.0/grid/rootupgrade -force
```
- Verify that the upgrade is complete by using the `crsctl query crs activeversion` command (as shown above).



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

If some nodes become unreachable in the middle of an upgrade, then you cannot complete the upgrade, because the upgrade script (`rootupgrade.sh`) did not run on the unreachable nodes. Because the upgrade is incomplete, Oracle Clusterware remains in the previous release. You can confirm that the upgrade is incomplete by entering the `crsctl query crs activeversion` command. To resolve this problem, run the `rootupgrade.sh` command with the `-force` flag by using the following syntax:

`<Grid_home>/rootupgrade -force`

Example:

```
# /u01/app/12.2.0/grid/rootupgrade -force
```

This command forces the upgrade to complete. Verify that the upgrade has completed by using the `crsctl query crs activeversion` command. The active release should be the upgrade release. The force cluster upgrade command has the following limitations:

- All active nodes must be upgraded to the newer release.
- All inactive nodes (accessible or inaccessible) may be either upgraded or not upgraded.
- For inaccessible nodes, after patch set upgrades, you can delete the node from the cluster. If the node becomes accessible later, and the patch version upgrade path is supported, then you can upgrade it to the new patch version.

Deinstalling the Old Oracle Clusterware Installation

- As root, change the permission and ownership of the previous release Grid home:

```
# chown -R grid:oinstall /u01/app/11.2.0/grid
# chmod -R 755 /u01/app/11.2.0/grid
```

- Log in as Installation owner and use the Grid Infrastructure deinstallation tool to remove the old Grid home.

```
$ cd /u01/app/11.2.0/grid/deinstall
$ ./deinstall
```

- You can obtain the stand-alone deinstallation tool from the following URL:

- <http://www.oracle.com/technetwork/database/enterprise-edition/downloads>



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

After upgrade from previous releases, if you want to deinstall the previous release Oracle Grid Infrastructure Grid home, then you must first change the permission and ownership of the previous release Grid home. Complete this task using the following procedure:

- Log in as root, and change the permission and ownership of the previous release Grid home by using the following command syntax, where *oldGH* is the previous release Grid home, *swowner* is the Oracle Grid Infrastructure installation owner:

```
# chown -R swowner oldGH
# chmod -R 755 oldGH
```

- After you change the permissions and ownership of the previous release Grid home, log in as the Oracle Grid Infrastructure Installation owner, and use the Oracle Grid Infrastructure stand-alone deinstallation tool to remove the previous release Grid home (*oldGH*).

Note: You must use the deinstallation tool from the same release to remove Oracle software. Do not run the deinstallation tool from a later release to remove Oracle software from an earlier release. For example, do not run the deinstallation tool from the 12.1.0.1 installation media to remove Oracle software from an existing 11.2.0.4 Oracle home.

- Obtain the stand-alone deinstallation tool with the URL shown in the slide.

Click the See All link for the downloads for your operating system platform, and scan the list of downloads for the *deinstall* utility.

Patch and Patch Set: Overview

- For its software, Oracle Corporation issues product fixes called *patches*.
- Patches are associated with particular releases and versions of Oracle products.
- The patching cycle involves downloading patches, applying patches, and verifying the applied patch.
- Patching involves migrating from one version of the software product to another within a particular release.
- When a patch is applied to an Oracle software installation, it updates the executable files, libraries, and object files in the software home directory.
 - The patch application can also update configuration files and Oracle-supplied SQL schemas.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Types of Patches

| Patch Type | Description |
|---|--|
| Interim Patches | Released to fix a bug, or a collection of bugs. Previously called patch set exceptions (PSE), one-off patches, or hot fixes. |
| Interim Patches (for Security bug fixes) | Released to provide customer-specific security fixes. Previously referred to as a test patch, fix verification binary, or e-fix. |
| Diagnostic Patches | Mainly help diagnose and verify a fix, or a collection of bugfixes |
| Bundle Patch Updates | Cumulative collection of fixes for a specific product or component. Previously referred to as a maintenance pack, service pack, cumulative patch, update release, or MLR. |
| Patch Set Updates (PSU) | Cumulative patch bundles that contain well-tested and proven bug fixes for critical issues. PSUs have limited new content, and do not include any changes that require re-certification. |
| Security Patch Updates | A cumulative collection of security bug fixes. Previously known as Critical Patch Updates, or CPUs. |



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Interim patches are bug fixes available to customers in response to specific bugs. They require a particular base release or patch set to be installed before you can apply them. These patches are not versioned and are generally available in a future patch set as well as the next product release. Interim patches are applied by using Enterprise Manager Cloud Control or OPatch, which is included with your Oracle Database installation.

Patch sets updates (PSUs) and patch bundles are mechanisms for delivering fully tested and integrated product fixes. All the fixes in a patch set have been tested and are certified to work with each other. Because a patch set includes only low impact patches, it does not require you to certify applications or tools against the updated Oracle Database software. When you apply a patch set, many different files and utilities are modified. This results in a release number change for your Oracle software. You use OPatch to apply PSUs and Oracle Universal Installer (OUI) to install patch sets. PSUs are rolling installable but patch sets are not.

Obtaining Oracle RAC Patch Sets

The screenshot shows the Oracle My Oracle Support interface. The top navigation bar includes links for Dashboard, Knowledge, Service Requests, Patches & Updates (which is selected), Community, Certifications, Managed Cloud, CRM On Demand, and Sy. Below the navigation is a search bar with tabs for Patching Quick Links and Patch Search. The main content area is titled 'Patches and Updates' and features sections for Software and Patch Search Sites, Oracle E-Business Suite, and Oracle Server and Tools. Under Oracle Server and Tools, there is a link to 'Latest Patchsets'. A red box highlights this link, and a red arrow points from it to the 'Patch Sets for Product Bundles' section. This section lists various Oracle products with their corresponding patch sets. The 'Oracle Database' row is also highlighted with a red box. The bottom of the page includes an Oracle logo and a copyright notice: 'Copyright © 2018, Oracle and/or its affiliates. All rights reserved.'

The latest patch sets and recommended BPs can be downloaded from the My Oracle Support website at the following URL:

<http://support.oracle.com/>

After signing in to the website, click the “Patches & Updates” tab. For the latest patch sets and patch bundles, click the “Latest Patchsets” link under Oracle Servers and Tools. You can choose from patch sets for product bundles or patch bundles for individual products.

If you know the patch set number, you can click the Number/Name Sun CR ID link. You can enter a single patch set number or a comma-separated list. Select your platform and click the Search button.

To locate the Patch Set notes on My Oracle Support:

1. Log in to My Oracle Support.
2. Select the **Patches & Updates** tab.
3. Select **Quick Links to the Latest Patchsets, Mini Packs, and Maintenance Packs**.
4. Under the heading Latest Oracle Server/Tools Patchsets, select **Oracle Database**.
A list of operating systems appears.
5. Place your cursor over the entry that matches your operating system, or use the triangular arrows to search for your operating system.
When you place the cursor over the entry for your operating system, for example, Linux x86, a list of database versions appears.
6. Select **12.2.0**
The Advanced Search page appears.
7. Scroll to the bottom of this page to see the list of available patch sets.
8. Select the number in the Patch column for the patch set you want to view or download.
The Patchset description and download page appears.
9. Click **View Readme** to see the patch set notes.
On this page you can also click **Download** to download the patch to your computer.
10. If you choose to download the patch, then follow the instructions in the ReadMe file of the patch set to apply the patch set to your software.

Obtaining Oracle Clusterware Patches

The screenshot shows two windows related to Oracle patch management:

- Top Window (Patch Search):** A search interface where filters are applied: Product (Oracle Clusterware), Release (12.2.0.1.0), and Platform (Linux x86-64). A red arrow points from this window to the main search results page below.
- Bottom Window (Patch Search Results):** The search results page titled "Patch Search" under "Patches & Updates". It displays a table of recommended patches:

| Patch Name | Description | Release | Platform (Language) | Recommended | Classification | Product |
|------------|--|------------|---------------------------------|-------------|----------------|--|
| 26878187 | GRID INFRASTRUCTURE RELEASE UPDATE REVISION 12.2.0.1.171017 (System Patch) | 12.2.0.1.0 | Linux x86-64 (American English) | Yes | Security | Oracle Database - Enterprise Edition (More...) |
| 26737266 | GRID INFRASTRUCTURE RELEASE UPDATE 12.2.0.1.171017 (System Patch) | 12.2.0.1.0 | Linux x86-64 (American English) | Yes | Security | Oracle Database - Enterprise Edition (More...) |

Both windows include an "ORACLE" logo at the bottom.

You obtain patches and patch sets from My Oracle Support, which is the Oracle Support Services website. The Oracle Support Services website is located at: <https://support.oracle.com>

To locate patches on the My Oracle Support website:

1. Log in to your account on My Oracle Support.
2. Select the **Patches & Updates** tab.
3. If you know the patch number, then you can enter it into the Patch Name or Number field, then click **Search**.

If you want to search for all available patches for your system, then select **Product or Family (Advanced Search)**, which is located above the Patch Name or Number field.

Supply the following information:

- Choose the products you want to patch (for example, Oracle Clusterware, Oracle Database, or an individual product such as Universal Installer)
- Specify the software release for the products you selected, for example, Oracle 12.2.0.1.0.
- Specify the platform on which the software is installed.

Click Search to look for available patches. The Patch Search Results page is displayed.

4. On the Patch Search Results page, select the number of the patch you want to download. A details page for that patch appears on your screen.
5. Click the **ReadMe** button to view the ReadMe file for the patch, or click **Download** to download the patch to your local computer.

Note: To locate recommended patches, start from the “Patches & Updates” tab and click the Recommended Patch Advisor link. Select Oracle Clusterware from the Product pull-down menu, and then select the release number and platform. Click the Search button for a list of recommended patches

Downloading Patches

The screenshot illustrates the Oracle Patch Advisor interface for downloading patches. It consists of three main windows:

- Patch Search Window:** Shows a list of patches. The patch "26737266" is selected and highlighted with a red box. A red arrow points from this selection to the "Patch Search" window below.
- Patch Summary Window:** Displays detailed information for "Patch 26737266: GRID INFRASTRUCTURE RELEASE UPDATE 12.2.0.1.171017". It includes fields for Product (Oracle Database - Enterprise Edition), Last Updated (17-Oct-2017 07:53 (1+ month ago)), Size (1010.0 MB), and Release (Oracle Database 12.2.0.1.0). The "Download" button is highlighted with a red box.
- File Download Window:** Shows the file "GRID INFRASTRUCTURE RELEASE UPDATE 12.2.0.1.171017 (Patch) p26737266_122010_Linux-x86-64.zip" listed for download. The "Download" button is also highlighted with a red box. A red arrow points from the "Download" button in the summary window to this download link.

Once you have located the patch you need, you can download it. Click the patch link on the search results page. Locate and click the Download link on the patch summary page, and then click the patch link in the File Download dialog box. Click the Save File button, and then click OK. Specify the directory location for the file, and then click Save.

Grid Infrastructure Patching methods

- OUI Installs *patch sets* as out-of-place upgrades, reducing the down time required for patching.
- OPatch supports 3 patch methods on a RAC environment
 - *All-Node Patch*: all the nodes in the cluster are initially shut down and the patch is applied on all the nodes.
 - *Minimum Downtime Patch*: Minimum downtime patching shortens the time that all the nodes have to be down to apply a patch.
 - *Rolling Patch*: This is the most efficient means of applying an interim patch to an Oracle RAC or Oracle Grid Infrastructure for a cluster installation. By patching groups of nodes individually, there is zero downtime for the cluster database because at least one instance is always available on a different node.

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Patch sets are 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.

OPatch supports 3 different patch methods on an Oracle cluster environment:

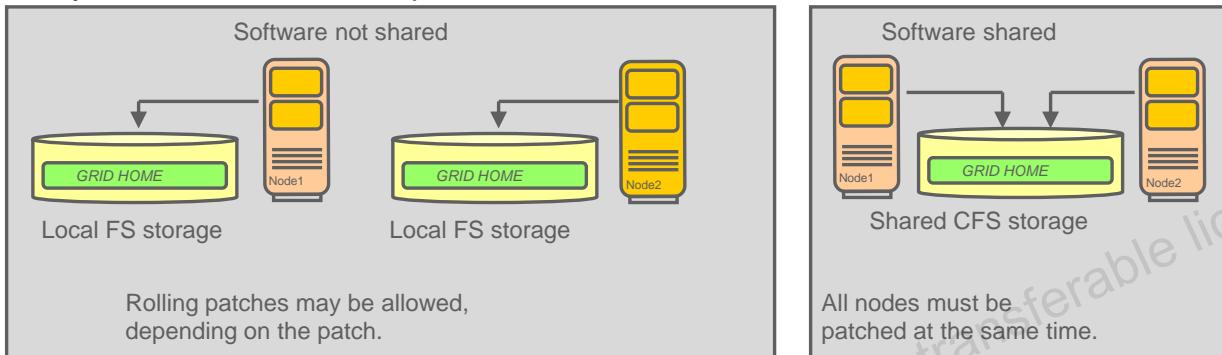
- **All-Node Patch**: In all node patching, all the nodes in the cluster are initially shut down and the patch is applied on all the nodes. After all the nodes have been patched, then the Oracle Clusterware stack and all its registered resources are restarted on each node. This method is typically used for very critical patches and it leads to maximum downtime. OPatch uses this method if the patch cannot be applied in a rolling fashion, and you did not specify the `minimize_downtime` option.
- **Minimum Downtime Patch**: Minimum downtime patching shortens the time that all the nodes have to be down to apply a patch. In minimum downtime patching, 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 downtime for Oracle RAC, compared to having all the nodes shut down at the same time.

- **Rolling Patch:** Rolling patching is performed group by group, separately, until all the nodes in the cluster are patched. This is the most efficient means of applying an interim patch to an Oracle RAC or Oracle Grid Infrastructure for a cluster installation. By patching groups of nodes individually, there is zero downtime for the cluster database because at least one instance is always available on a different node.
While most patches can be applied in a rolling fashion, some patches cannot be applied in this fashion. The README file for the patch indicates whether you can apply the patch using the rolling patch method. If the patch cannot be applied using the rolling patch method, then you must use either “Minimum Downtime Patching” or “All Node Patching” to apply the patch.

Rolling Patches

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

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



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

In rolling patching, one group of nodes is shut down, the patch is applied to those nodes, and the nodes are brought back up. This is performed group by group, separately, until all the nodes in the cluster are patched. This is the most efficient means of applying an interim patch to an Oracle RAC or Oracle Grid Infrastructure for a cluster installation. By patching groups of nodes individually, there is zero downtime for the cluster database because at least one instance is available at all times on a different node.

Although most patches can be applied in a rolling fashion, some patches cannot be applied in this fashion. The README file for the patch indicates whether or not you can apply the patch by using the rolling patch method. If the patch cannot be applied using the rolling patch method, then you must use either “Minimum Downtime Patching” or “All Node Patching” to apply the patch.

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.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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 [12.1.0.2.0]
```

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 [12.1.0.1.0]
```

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 a later version than those running other Oracle products such as the Real Application Clusters (RAC) database and ASM software versions.

OPatch: Overview

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



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

OPatch is an Oracle-supplied utility that assists you with the process of applying interim patches to Oracle's software. Opatch is a Java-based utility that can run on either OUI-based Oracle homes or stand-alone homes. It works on all operating systems for which Oracle releases software.

OPatch is included with the Oracle Clusterware 12c 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.

OPatch: General Usage

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

```
$ export ORACLE_HOME=/u01/app/12.2.0/grid
$ opatch command [options]
```

or

```
$ opatch command -oh /u01/app/12.2.0/grid [options]
```

- To obtain help with the OPatch syntax:

```
$ opatch command -help
```

- To check whether a patch supports a rolling application (run it from the `patch` directory):

```
$ opatch query query -is_rolling_patch <patch_location>
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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. 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.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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 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 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 Patch Manually Using OPatch

- Verify that Oracle Inventory is properly configured.

```
[grid]$ opatch lsinventory
```

- Unlock the protected files as root user.

```
[root]# $GRID_HOME/crs/install/rootcrs.pl -prepatch
```

- Patch the CRS installation on the first node only.

```
[grid]$ opatch apply -oh $GRID_HOME -local <patch_location>
```

- Lock the protected files as root user.

```
[root]# $GRID_HOME/crs/install/rootcrs.pl -postpatch
```

- Verify patch installation.

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

- Repeat steps 1–5 on each node, one at a time.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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. The example in the slide shows how to apply a clusterware patch manually. The following example shows how to apply a PSU patch manually.

- Check for conflicts by running the following command.

```
[root]# $GRID_HOME/Opatch/opatchauto apply <patch_location> -apply
```

- Verifying that Oracle Inventory can be located and properly configured with the following command. If the ORACLE_HOME environment variable has been defined, it is not necessary to include the -oh option.

```
[grid]$ opatch lsinventory -detail -oh <GRID_HOME>
```

- Stop all databases and services running out of the RDBMS_HOME.

```
[oracle]$ srvctl stop instance -db orcl -instance orcl_1
```

```
[oracle]$ srvctl stop home -o $RDBMS_HOME -s /tmp/status -n host01
```

- Run the pre-root script to unlock the protected files.

```
[root]# $GRID_HOME/crs/install/rootcrs.pl -prepatch
```

5. Patch the CRS installation on the first node.
[grid]\$ \$GRID_HOME/OPatch/opatch apply -oh \$GRID_HOME -local
<patch_location>
6. Patch the DB installation on the first node.
[oracle]\$ \$ORACLE_HOME/OPatch/opatch apply -oh \$ORACLE_HOME -local
<patch_location>
7. Run the post script.
[root]\$ \$GRID_HOME/rdbms/install/rootadd_rdbms.sh
[root]\$ \$GRID_HOME/crs/install/rootcrs.pl -postpatch
8. Verify patch installation.
[grid]\$ opatch lsinventory -detail -oh <GRID_HOME>
9. Repeat all steps above for the remaining nodes.
10. Run the datapatch tool for each Oracle database.
[oracle]\$ \$ORACLE_HOME/OPatch/datapatch -verbose

OPatch Automation

- OPatch has automated patch application for the Oracle Grid Infrastructure and Oracle RAC database homes.
- Existing configurations are queried. The steps that are required for patching each Oracle RAC database home of the same version and the Grid home are automated.
- The utility must be executed by an operating system user with root privileges.
- OPatch must be executed on each node in the cluster if the Grid home or RAC home is in nonshared storage.
- One invocation of OPatch can patch the Grid home, one or more RAC homes, or both Grid and Oracle RAC database homes of the same Oracle release version.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Installing a Patch Automatically Using OPatchAuto

- To patch Grid home and all Oracle RAC database homes of the same version:

```
# opatchauto apply <UNZIPPED_PATCH_LOCATION> <Grid_home> -ocmrf
<ocm_response_file>
```

- To patch only the GI home:

```
# opatchauto apply <UNZIPPED_PATCH_LOCATION> -oh <Grid_home> -ocmrf
<ocm_response_file>
```

- To patch one or more Oracle RAC database homes:

```
# opatchauto apply <UNZIPPED_PATCH_LOCATION> -database db1, db2 -ocmrf
<ocm_response_file>
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

If you have not installed Oracle Configuration Manager (OCM), the OPatch utility will prompt for your OCM response file when it is run. You should enter a complete path of OCM response file if you already have created this in your environment. If you do not have the OCM response file (`ocm.rsp`), you should run the `emocmrsp` command to create it. As the software home owner, execute:

```
$ <ORACLE_HOME>/OPatch/ocm/bin/emocmrsp
```

Before executing OPatch, add the directory containing OPatch to the your path:

```
# export PATH=$PATH:<GI_HOME>/Opatch
```

To patch GI home and all Oracle RAC database homes of the same version:

```
# <Grid_home>/OPatch/opatchauto apply <Grid_home> -ocmrf
<ocm_response_file>
```

To patch only the GI home:

```
# <Grid_home>/OPatch/opatchauto apply <SYSTEM_PATCH_TOP_DIR> -oh
<Grid_home> -ocmrf <ocm_response_file>
```

To patch one or more Oracle RAC databases and associated GI/RAC homes:

```
# opatchauto apply <UNZIPPED_PATCH_LOCATION> -database db1, db2 -ocmrf
<ocm_response_file>
```

To roll back the patch from the GI home and each Oracle RAC database home:

```
# opatchauto rollback <UNZIPPED_PATCH_LOCATION> -ocmrf <ocm_response_file>
```

To roll back the patch from the GI home:

```
# opatchauto rollback <UNZIPPED_PATCH_LOCATION> -oh <Grid_home>      -ocmrf  
<ocm_response_file>
```

To roll back the patch from one or more Oracle RAC database homes:

```
# opatchauto rollback <UNZIPPED_PATCH_LOCATION> -database db1, db2 -ocmrf  
<ocm_response_file>
```

OPatch Log and Trace Files

- OPatch maintains logs for apply, rollback, and lsinventory operations.
- OPatch Log files are located in `ORACLE_HOME/cfgtoollogs/opatch`.
- Each log file is tagged with the time stamp of the operation.
- Each time you run OPatch, a new log file is created.
- OPatch maintains an index of processed commands and log files in the `opatch_history.txt` file.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Logging and tracing is a common aid in debugging. OPatch maintains logs for apply, rollback, and lsinventory operations. Log files are located in `Oracle_home/cfgtoollogs/opatch`. Each log file is tagged with the time stamp of the operation. Log files are named as `opatch_mm-dd-yyyy_hh-mm-ss.log`, where `mm-dd-yyyy` is the current date and `hh-mm-ss` is the current time. Each time you run OPatch, a new log file is created.

For example, if a log file is created on May 17, 2013 at 11:55 PM, then it is named as follows:

`opatch_05-17-2013_23-55-00.log`

OPatch also maintains an index of the commands processed by OPatch and the log files associated with it in the `opatch_history.txt` file located in the `Oracle_home/cfgtoollogs/opatch` directory. A sample of the `opatch_history.txt` file is as follows:

```
Date & Time : Tue Apr 26 23:00:55 PDT 2013
Oracle Home : /u01/app/oracle/product/12.1.0/dbhome_1/
OPatch Ver. : 12.1.0.0.0
Current Dir : /scratch/oui/OPatch
Command : lsinventory
Log File : /u01/app/oracle/product/12.1.0/dbhome_1/cfgtoollogs/opatch/opatch-
2013_Apr_26_23-00-55-PDT_Tue.log
```

Queryable Patch Inventory

- The `DBMS_QOPATCH` package provides a PL/SQL or SQL interface to view the database patches that are installed.
- The interface provides all the patch information that is available as part of the OPatch `lsinventory -xml` command.
- The package accesses the OUI patch inventory in real time to provide patch and patch meta information.
- The `DBMS_QOPATCH` package allows users to:
 - Query which patches are installed from SQL*Plus
 - Write wrapper programs to create reports and do validation checks across multiple environments
 - Check patches that are installed on cluster nodes from a single location



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Using `DBMS_QOPATCH`, Oracle Database 12c provides a PL/SQL or SQL interface to view the database patches that are installed. The interface provides all the patch information available as part of the OPatch `lsinventory -xml` command. The package accesses the Oracle Universal Installer (OUI) patch inventory in real time to provide patch and patch meta information. Using this feature, users can:

- Query what patches are installed from SQL*Plus
- Write wrapper programs to create reports and do validation checks across multiple environments
- Check patches installed on cluster nodes from a single location instead of having to log onto each one in turn

Alternative Methods of Patching

- Using Oracle Enterprise Manager Cloud Control for Patching Operations
 - Using Cloud Control with its Provisioning & Patching functionality, you can automate the patching of your Oracle Grid Infrastructure and Oracle RAC software.
- Rapid Home Provisioning, Patching, and Upgrading
 - Rapid Home Provisioning is a method of deploying software homes to any number of nodes in a data center from a single cluster, and also facilitates patching and upgrading software.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

OPatch is a commonly used method for patching Oracle software homes, but this is not the only method of patching Oracle software.

Here are two possible alternative methods of patching. You might find these methods more appropriate for your organization than using OPatch.

1. Oracle Enterprise Manager Cloud Control for Patching Operations
 - Using Cloud Control with its Provisioning & Patching functionality, you can automate the patching of your Oracle Grid Infrastructure and Oracle RAC software.
 - Details on how to patch your Oracle Grid Infrastructure and Oracle RAC software using Cloud Control are available from the following PDF file: <http://www.oracle.com/technetwork/oem/pdf/512066.pdf>
2. Rapid Home Provisioning, Patching, and Upgrading
 - Rapid Home Provisioning is a method of deploying software homes to any number of nodes in a data center from a single cluster, and also facilitates patching and upgrading software.
 - See Also: [Oracle Clusterware Administration and Deployment Guide](#)



Quiz

Which tools can be used to install a patchset?

- a. Oracle Universal Installer
- b. OPatch
- c. Enterprise Manager
- d. Database Configuration Assistant
- e. Rapid Home Provisioning



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.



Quiz

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

- a. True
- b. False



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Answer: a

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 lesson, you should have learned how to:

- Describe the types of patches and upgrades available
- Plan for rolling patches and rolling upgrades
- Compare software versions with the active version
- Install a patch with the `opatch` utility



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Monitoring and Troubleshooting Oracle Clusterware



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe the functions of the Cluster Verify Utility
- Describe the functions of the Cluster Health Monitor
- Describe the functions of the Cluster Health Advisor
- Describe the functions of the Trace File Analyzer Collector
- Use the Cluster Resource Activity Log
- Locate the Oracle Clusterware log files
- Troubleshoot Node Eviction



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Lesson Agenda

- Using Oracle Autonomous Health Framework
 - Cluster Verify Utility (CVU)
 - Cluster Health Monitor (CHM)
 - Cluster Health Advisor (CHA)
 - Trace File Analyzer (TFA) Collector
- Using the Cluster Resource Activity Log (CALOG)
- Using Oracle Clusterware Diagnostic and Alert Log Data
- Node Eviction



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

“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/ntp.pid"
```

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



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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, 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.

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

```
OPTIONS="-x -u ntp:ntp -p /var/run/ntp.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 Autonomous Health Framework

- Autonomous Health Framework is a set of utilities that collect and analyze diagnostic data, proactively identifying issues.
- Most of the Autonomous Health Framework components are already available in Oracle Database 12.1.
- In Oracle Database 12.2, the output of several components are consolidated in the Grid Infrastructure Management Repository (GIMR) and analyzed in real time.
- Autonomous Health Framework assists with monitoring, diagnosing, and preventing availability and performance issues.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Autonomous Health Framework is a collection of components that analyze the diagnostic data collected, and proactively identify issues before they affect the health of your clusters or Oracle RAC databases.

Most of the Oracle Autonomous Health Framework components are already available in Oracle Database 12c release 1. In Oracle Database 12c release 2, the output of several components is consolidated in the Grid Infrastructure Management Repository (GIMR) and analyzed in real time to detect problematic patterns on the production clusters.

Oracle Autonomous Health Framework assists with monitoring, diagnosing, and preventing availability and performance issues. System administrators can use most of the components in Oracle Autonomous Health Framework interactively during installation, patching, and upgrading. Database administrators can use Oracle Autonomous Health Framework to diagnose operational runtime issues and mitigate the impact of these issues.

Oracle Autonomous Health Framework components include:

- **Cluster Verification Utility (CVU):** The Cluster Verification Utility (CVU) can assist you in diagnosing a wide variety of configuration problems. As part of Autonomous Health Framework, CVU can perform these checks periodically by running autonomously on a scheduled basis.
- **Cluster Health Monitor (CHM):** CHM is a component of Grid Infrastructure, which constantly monitors and stores Oracle Clusterware and operating system resources metrics.
- **Oracle Cluster Health Advisor (CHA):** CHA, introduced in Oracle Database 12.2, continuously monitors cluster nodes and Oracle RAC databases for performance and availability issue precursors to provide early warnings of problems before they become critical.
- **Oracle Trace File Analyzer (TFA) Collector:** This is a utility for targeted diagnostic collection that simplifies diagnostic data collection for Oracle Clusterware, Oracle Grid Infrastructure, and Oracle RAC systems, in addition to single-instance, non-clustered databases.
- **Oracle ORAchk and Oracle EXAchk:** These provide a lightweight and nonintrusive health check framework for the Oracle stack of software and hardware components. ORAchk and EXAchk are available as value add-ons to your existing support contract. No additional fee or license is required to run Oracle ORAchk and Oracle EXAchk.
- **Memory Guard:** Memory Guard is an Oracle RAC environment feature to monitor cluster nodes to prevent node stress caused by the lack of memory.
- **Hang Manager:** In an Oracle RAC environment, the Hang Manager feature autonomously resolves hangs and keeps the resources available. Hang Manager was first available in Oracle Database 11.1 but has been improved for release 12.2.
The new features of Hang Manager are covered in the lesson titled “Oracle RAC New Features.”
- **Oracle Database Quality of Service (QoS) Management:** QoS manages the resources that are shared across applications.

Cluster Verify Utility (CVU)



- CVU performs Oracle clusterware health checks periodically by running autonomously on a scheduled basis.
- CVU verifies 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®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Clusterware resource (ora.cvud)

- ora.cvud invokes Clusterware Health Checks every 6 hrs.
- ora.cvud results are stored in
 - \$ORACLE_BASE/crsdata/@global/cvud/baseline/cvures
- To start/stop the ora.cvud resource.

```
$ srvctl start[stop] cvud
```

- To check the status of the ora.cvud resource.

```
$ srvctl status cvud
$ crsctl stat res ora.cvud -t
```

- To enable/disable the ora.cvud resource.

```
$ srvctl enable[disable] cvud
```

- To modify the current setting of the ora.cvud resource.

```
$ srvctl modify cvud [options]
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

ora.cvud is an Oracle clusterware resource that invokes clusterware health checks a regular interval (every 6 hrs by default). Starting with Oracle Grid Infrastructure 12.2, you can find the CVU results in \$ORACLE_BASE/crsdata/@global/cvud/baseline/cvures.

The `srvctl` utility can be used to control the ora.cvud resource.

- To check the status of CVU (ora.cvud)

```
[grid@host01 ~]$ crsctl stat res ora.cvud -t
```

| Name | Target | State | Server | State details |
|-------------------------------|--------|--------|--------|---------------|
| <hr/> Cluster Resources <hr/> | | | | |
| ora.cvud | 1 | ONLINE | ONLINE | host01 |
| | | | | STABLE |
| <hr/> | | | | |

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

- To check the status of CVU (ora.cvu)

```
[grid@host01 ~]$ srvctl status cvu
```

CVU is enabled and running on node host01

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

- To check the current interval setting of CVU. (21600 sec, every 6 hrs)

```
[grid@host01 ~]$ crsctl stat res ora.cvu -p | grep RUN_INTERVAL
```

RUN_INTERVAL=21600

- Attempt to modify the current interval setting using the crsctl utility.

```
[grid@host01 ~]$ crsctl modify resource ora.cvu -attr "RUN_INTERVAL=86400"
```

CRS-4995: The command 'Modify resource' is invalid in crsctl. Use srvctl for this command.

- Modify the current interval setting of CVU. (1440min, every 24 hrs)

```
[grid@host01 ~]$ srvctl modify cvu -help
```

Modifies the check interval for the CVU resource.

Usage: **srvctl modify cvu [-checkinterval <check_interval_in_minutes>] [-destloc <path>]**

-checkinterval <check_interval_in_minutes> Interval in minutes between checks

-destloc <path> Directory for copying and executing CVU files

-help Print usage

```
[grid@host01 ~]$ srvctl modify cvu -checkinterval 1440
```

```
[grid@host01 ~]$ srvctl config cvu
```

CVU is configured to run once every 1440 minutes

CVU is enabled.

CVU is individually enabled on nodes:

CVU is individually disabled on nodes:

CVU Health Check Report: Example

```
*****  
Summary of environment  
*****  
  
Date (mm/dd/yyyy)      : 11/13/2017  
Time (hh:mm:ss)        : 16:56:22  
Cluster name           : cluster01  
Clusterware version    : 12.2.0.1.0  
Grid home              : /u01/app/12.2.0/grid  
Grid User              : grid  
Operating system       : Linux4.1.12-94.3.6.el6uek.x86_64  
Database[el]            : Database name - orcl  
                           Database version - 12.2.0.1.0  
                           Database home -  
                           /u01/app/oracle/product/12.  
                           2.0/dbhome_1  
  
*****  
System requirements  
*****  
  
Verification Check     : Physical Memory  
Verification Description: This is a prerequisite condition to test whether the  
                           system has at least 8GB (8388608.0KB) of total physical  
                           memory.  
Verification Result    : PASSED  
Verification Summary   : Physical memory meets or exceeds recommendation
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

ora.cvu invokes the cluster verification utility. It executes the cluvfy comp health -_format command in the background. The example in the slide is a CVU result found in \$ORACLE_BASE/crsdata/@global/cvu/baseline/cvures.

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
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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 |
| nodecon | Checks node connectivity |
| cfs | Checks CFS integrity |
| ssa | Checks shared storage |
| space | Checks space availability |
| sys | Checks minimum requirements |
| clu | Checks cluster integrity |
| clumgr | Checks cluster manager integrity |
| ocr | Checks OCR integrity |
| admprv | Checks administrative privileges |
| software | Checks software distribution |
| clocksync | Checks clock synchronization |
| nodeapp | Checks node app existence |
| freespace | Checks free space in CRS Home |
| healthcheck | Checks mandatory requirements |

| | |
|-----------------|----------------------------------|
| peer | Compares properties with peers |
| ha | Checks HA integrity |
| asm | Checks ASM integrity |
| acfs | Checks ACFS integrity |
| olr | Checks OLR integrity |
| gpnp | Checks GPnP integrity |
| gns | Checks GNS integrity |
| scan | Checks SCAN configuration |
| ohasd | Checks OHASD integrity |
| crs | Checks CRS integrity |
| vdisk | Checks Voting Disk Udev settings |
| dhcp | Checks DHCP configuration |
| dns | Checks DNS configuration |
| baseline | Collects and compares baselines |

Cluster Verify Output: Example

```
$ cluvfy comp healthcheck -bestpractice -html -save
```

| System recommendations | | |
|--|---------------------|--|
| Verification Check | Verification Result | Verification Description |
| Ethernet Jumbo Frames | NOT MET | Checks if Jumbo Frames are configured on the system... details |
| HugePages Existence | NOT MET | Checks HugePages existence |
| Hardware Clock synchronization at shutdown | MET | Checks whether Hardware Clock is synchronized with the system clock during system shutdown |
| availability of port 8888 | MET | availability of port 8888 |

| Clusterware recommendations | | |
|------------------------------------|---------------------|---|
| Verification Check | Verification Result | Verification Description |
| CSS misscount parameter | MET | Checks if the CSS misscount is set correctly on the system... details |
| CSS reboottime parameter | MET | Checks if the CSS reboottime is set correctly on the system... details |
| CSS disktimeout parameter | MET | Checks if the CSS disktimeout is set correctly on the system... details |
| ACFS device special file | MET | checks the attributes for the ACFS device special file... details |



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The slide shows you the output of the `cluvfy comp healthcheck -bestpractice -html -save` command. The result file is saved in `/u01/app/grid/crsdata/@global/cvu/report/html/cvucheckreport_<timestamp>.html`

This command checks that the mandatory requirements and/or best practice recommendations are met for successful operation of Oracle Clusterware, ASM and the database(s) configured on the system. The checks related to Oracle Clusterware range from the operating system-specific software requirements to networking or storage system configurations to the Oracle Clusterware specific components like OCR, CRS, and SCAN to name a few. The scope of validation is specified through the combination of command line options `-cluster`, `-database`, `-asm`, `-bestpractice`, and `-mandatory`. If the request has been made to perform validations for the databases but no database has been explicitly specified in the command line, then all the databases configured on the system are discovered and validations are performed for each discovered database. The results of the validation can be displayed as text or as HTML (if feasible). The detailed report of results can be requested to be saved for a future reference.

Cluster Health Monitor (CHM)



- CHM stores O/S metrics in the CHM repository that can be used for troubleshooting cluster issues.
- CHM consists of the following services:
 - `osysmond`: System Monitor Service, one per node
 - `ologgerd`: Cluster Logger Service, one primary and one standby per cluster
- If the master cluster logger service fails, the node where the standby resides:
 - Takes over as master
 - Selects a new node for standby
- The CHM repository is managed by using the `oclumon` utility.

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

oclumon Utility

- **oclumon** can be used to query the CHM repository to display node-specific metrics for a specified time period.
- **oclumon** can query and print the durations and the states for a resource on a node during a specified time period.
 - These states are based on predefined thresholds for each metric and are denoted as red, orange, yellow, and green.
- The **oclumon** utility can be run interactively, or with one of the following verbs:
 - dumpnodeview
 - manage
 - version
 - debug
 - analyze



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The **oclumon** command-line tool is included with CHM and you can use it to query the CHM repository to display node-specific metrics for a specified time period. You can also use **oclumon** to query and print the durations and the states for a resource on a node during a specified time period. These states are based on predefined thresholds for each resource metric and are denoted as red, orange, yellow, and green, indicating a decreasing order of criticality. For example, you can query to show how many seconds the CPU on a node named node1 remained in the RED state during the last hour. You can also use OCLUMON to perform miscellaneous administrative tasks, such as changing the debug levels, querying the version of CHM, and changing the metrics database size.

The **oclumon** utility can be run interactively, or with one of the following **oclumon** commands **dumpnodeview**, **manage**, **version**, **analyze**, and **debug**.

```
$ oclumon
```

```
query> help
For help from command line  : oclumon <verb> -h
For help in interactive mode : <verb> -h
Currently supported verbs are  :
dumpnodeview, manage, version, debug, analyze, quit, exit, and help
```

clomon dumpnodeview Command

- The `clomon dumpnodeview` command displays log information from `osysmond` in the form of a node view.
- A node view is a collection of all metrics collected by CHM for a node at a point in time.
- CHM attempts to collect metrics every second on every node.
- The syntax for the `clomon dumpnodeview` command:

```
clomon dumpnodeview [[-allnodes] | [-n node1 node2] [-last "duration"] | [-s "time_stamp" -e "time_stamp"] [-v] [-warning]] [-h]
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Use the `clomon dumpnodeview` command to view log information from the system monitor service in the form of a node view. A node view is a collection of all metrics collected by CHM for a node at a point in time. CHM attempts to collect metrics every second on every node. Some metrics are static whereas other metrics are dynamic. A node view consists of seven views when you display verbose output:

- **SYSTEM:** Lists system metrics such as CPU COUNT, CPU USAGE, and MEM USAGE
- **TOP CONSUMERS:** Lists the top consuming processes in the following format:
metric_name: '*process_name(process_identifier) utilization*'
- **PROCESSES:** Lists process metrics such as PID, name, number of threads, memory usage, and number of file descriptors
- **DEVICES:** Lists device metrics such as disk read and write rates, queue length, and wait time per I/O
- **NICS:** Lists network interface card metrics such as network receive and send rates, effective bandwidth, and error rates
- **FILESYSTEMS:** Lists file system metrics, such as total, used, and available space
- **PROTOCOL ERRORS:** Lists any protocol errors

oclumon dumpnodeview Command

- The following example dumps node views from node1, node2, and node3 collected over the last 12 hours:

```
$ oclumon dumpnodeview -n node1 node2 node3 -last "12:00:00"
```

- To display node views from all nodes collected over the last 15 minutes:

```
$ oclumon dumpnodeview -allnodes -last "00:15:00"
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Command parameters include:

- allnodes**: Use this option to dump the node views of all the nodes in the cluster.
- n node1 node2**: Specify one node (or several nodes in a space-delimited list) for which you want to dump the node view.
- last "duration"**: Use this option to specify a time, given in the HH24:MM:SS format surrounded by double quotation marks (""), to retrieve the last metrics (for example: "23:05:00.")
- s "time_stamp" -e "time_stamp"**: Use the -s option to specify a time stamp from which to start a range of queries and use the -e option to specify a time stamp to end the range of queries. Specify time in the YYYY-MM-DD HH24:MM:SS format surrounded by double quotation marks ("").

oclumon manage Command

- Use the oclumon manage command to view log information from the system monitor service.
- To display CHM repository properties:

```
$ oclumon manage -get repsize
CHM Repository Size = 90900 seconds

$ oclumon manage -get reppath
CHM Repository Path =
+DATA/_MGMTDB/FD9B43BF6A646F8CE043B6A9E80A2815/DATAFILE/sysmgmtdata.269.88
2743737

$ oclumon manage -get master
Master = host03
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Use the oclumon manage command to view log information from the system monitor service. This command can be used for various CHM management tasks including:

- Resizing the CHM repository to a specified window between one hour and three days
- Managing the CHM repository size by setting a space limit
- Changing the CHM repository location

The syntax for the oclumon manage command is:

```
oclumon manage [[-repos {resize_size | changesize memory_size |
reploc new_location [-maxtime size] | [-maxspace memory_size]}] | |
[-get key1 key2 ...]]
```

Note: Both the local system monitor service and the master cluster logger service must be running to resize the CHM repository.

Cluster Health Monitor (CHM) Enhancements

- Use the `-format csv` option for the `oclumon` command to output content in comma-separated values file format.

```
$ oclumon dumpnodeview -format csv
```

- Use the `-procag` option for `oclumon` to format node view processes aggregated by category.

```
$ oclumon dumpnodeview -procag
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Two new parameters have been added to the `oclumon dumpnodeview` command. You can use the `-format csv` option to format the command output into comma-separated values (CSV).

```
# oclumon dumpnodeview -format csv
dumpnodeview: Node name not given. Querying for the local host
-----
Node: host01 Clock: '2017-03-27 20.15.20+0000' SerialNo:89373
-----
SYSTEM:
"#pcpus", "#cores", "#vcpus", "cpuhrt", "chipname", "cpuusage[%]", "cpusys[%]", "cpuuse
r[%]", "cpunice[%]", "cpuiowait[%]", "cpusteal[%]", "cpuq", "physmemfree[KB]", "physm
emtotal[KB]", "mcache[KB]", "swapfree[KB]", "swaptotal[KB]", "hugepagetotal", ...
```

The `-procag` option for the `oclumon dumpnodeview` command provides output of node view processes aggregated by the following categories:

- DBBG (DB backgrounds)
- MDBG (GIMR backgrounds)
- ASMBG (ASM backgrounds)
- DBFG (DB foregrounds)

Cluster Health Advisor (CHA)

- CVU
- CHM
- **CHA**
- TFA

- Oracle Cluster Health Advisor continuously monitors cluster nodes and RAC databases for performance and availability issues.
 - Provides early-warning alerts and corrective action
 - Supports on-site calibration to improve sensitivity
- Oracle Cluster Health Advisor is integrated into Oracle Enterprise Manager Cloud Control (EMCC) Incident Manager.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Cluster Health Advisor is introduced for Oracle Database 12.2. It continuously monitors cluster nodes and RAC databases for performance and availability issue precursors to provide early warnings of problems before they become critical. Oracle Cluster Health Advisor is integrated into Oracle Enterprise Manager Cloud Control (EMCC) Incident Manager, and does the following:

- Detects node and database performance problems
- Provides early-warning alerts and corrective action
- Supports on-site calibration to improve sensitivity

In Oracle Database 12c release 2, Oracle Cluster Health Advisor supports monitoring of two critical subsystems of Oracle Real Application Clusters (Oracle RAC): the database instance and the host system. Oracle Cluster Health Advisor determines and tracks the health status of the monitored system. It periodically samples a wide variety of key measurements in the monitored system.

Oracle Cluster Health Advisor runs an analysis multiple times a minute. It estimates an expected value of an observed input based on the default model. It then performs anomaly detection for each input based on the difference between observed and expected values. If sufficient inputs associated with a specific problem are abnormal, it issues a warning and generates an immediate targeted diagnosis and corrective action.

Cluster Health Advisor: Overview

- Cluster Health Advisor models are conservative to prevent false warning notifications.
- It is possible that the default model configuration might not be sensitive enough for critical production systems.
 - CHA provides a model calibration capability to use production workload data to form the basis of its default setting.
- CHA stores the analysis results, diagnosis, corrective action, and metric evidence for later triage, in the GIMR.
- The capability of CHA to review past problems is managed by the retention setting for the CHA tablespace in the GIMR.
- The default retention period for the monitored data is 72 hours.



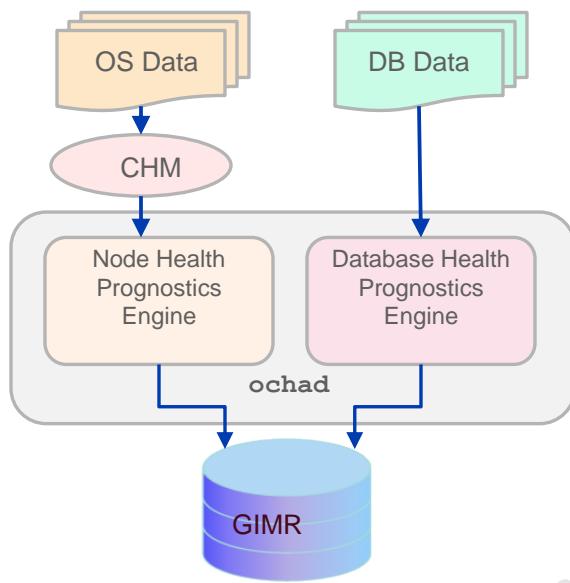
Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Cluster Health Advisor models are conservative to prevent false warning notifications. However, the default configuration may not be sensitive enough for critical production systems. Therefore, Oracle Cluster Health Advisor provides an on-site model calibration capability to use actual production workload data to form the basis of its default setting and to increase the accuracy and sensitivity of the node and database models.

Oracle Cluster Health Advisor stores the analysis results, along with diagnosis information, corrective action, and metric evidence for later triage, in the Grid Infrastructure Management Repository (GIMR). It also sends warning messages to Enterprise Manager Cloud Control by using the Oracle Clusterware event notification protocol.

You can also use Oracle Cluster Health Advisor to diagnose and triage past problems. You specify the past dates through Oracle Enterprise Manager Cloud Control (EMCC) Incident Manager or through the command-line interface CHACTL. The capability of Oracle Cluster Health Advisor to review past problems is managed by the retention setting for Oracle Cluster Health Advisor's tablespace in GIMR. The default retention period is 72 hours.

Oracle Cluster Health Advisor Architecture



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Cluster Health Advisor runs as a highly available cluster resource, OCHAD, on each node in the cluster. Each Oracle Cluster Health Advisor daemon (`ochad`) monitors the operating system on the cluster node and optionally, each Oracle Real Application Clusters (Oracle RAC) database instance on the node.

The `ochad` daemon receives operating system metric data from the Cluster Health Monitor and gets Oracle RAC database instance metrics from a memory-mapped file. The daemon does not require a connection to each database instance. This data, along with the selected model, is used in the Health Prognostics Engine of Oracle Cluster Health Advisor for both the node and each monitored database instance to analyze their health multiple times a minute.

Using the CHA Command Line Interface `chactl`

- Use `chactl monitor` to start monitoring all the instances of a specific Oracle RAC database by using the current set model.

```
$ chactl monitor database -db HRdb -model Mar2017
```

- Use the `chactl status` command to check the monitoring status of the running targets.

```
# chactl status
Monitoring nodes host01, host02
Monitoring databases SalesDB, HRdb
```

- Use the `chactl config` command to list all the targets being monitored, along with the current model of each target.

```
$ chactl config
Databases monitored: prodDB, hrDB
$ chactl config cluster
Monitor: Enabled
Model: DEFAULT_CLUSTER
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Use the `chactl monitor` command to start monitoring all the instances of a specific Oracle RAC database by using the current set model.

Use the `chactl status` command to check the monitoring status of the running targets. If you do not specify any parameters, the `chactl status` command returns the status of all running targets. The monitoring status of an Oracle Cluster Health Advisor target can be either Monitoring or Not Monitoring.

Use the `chactl config` command to list all the targets being monitored, along with the current model of each target. The `chactl config` command also displays the configuration data status if the specified target is a multitenant container database (CDB) or a cluster.

Managing the CHA Models: Defining “normal”

- Use `chactl calibrate` to create a new model that has greater sensitivity and accuracy.

```
$ chactl calibrate database -db orcl -model WEEKDAY -timeranges 'start=start=2017-02-09 16:00:00,end=2017-02-09 23:00:00'  
$ chactl calibrate cluster -model MY_CLUSTER -timeranges 'start=2017-03-22 13:50:00,end=2017-03-24 07:00:00'
```

- CHA successfully performs the calibration if 720 or more records are available.
- A set of metrics or Key Performance Indicators describe high-level constraints to the training data selected for calibration.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Use the `chactl calibrate` command to create a new model that has greater sensitivity and accuracy. The user-generated models are effective for the Oracle RAC monitored systems in your operating environment because the user-generated models use calibration data from the target. Oracle Cluster Health Advisor adds the user-generated model to the list of available models and stores the new model in the CHA repository.

CHA successfully performs the calibration if 720 or more records are available. The calibration function may not consider some data records to be normally occurring for the workload profile being used. In this case, filter the data by using the `KPISET` parameters in both the `query calibration` command and the `calibrate` command. Run the `query calibration` command to check if sufficient data is available.

```
$ chactl query calibration -db db_name -timeranges 'start=start_time\end=end_time' -kpiset 'name=CPUPERCENT min=20 max=40'
```

A set of metrics or Key Performance Indicators (`kpiset`) describe high-level constraints to the training data that is selected for calibration. This set consists of relevant metrics to describe performance goals and resource utilization bandwidth, for example, response times or CPU utilization.

CHA Key Performance and Workload Indicators

- Key Performance Indicators supported for database:
 - CPUPERCENT: CPU utilization as a percentage
 - IOREAD: Disk reads, Mbyte/sec
 - DBTIMEPERCALL: Database time per user call, usec/call
 - IOWRITE: Disk write, Mbyte/sec
 - IOTHROUGHPUT: Disk throughput, IO/sec

```
$ chactl calibrate database -db oracle -model weekday  
-timeranges 'start=start=2017-05-05 16:00:00,end=2017-05-05 23:00:00' -kpiset  
'name=CPUPERCENT min=10 max=60'
```

- Key Performance Indicators supported for cluster:

- CPUPERCENT: CPU utilization as a percentage
- IOREAD: Disk read, Mbyte/sec
- IOWRITE: Disk write, Mbyte/sec
- IOTHROUGHPUT: Disk throughput, IO/sec



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Using chactl query to View Problems and Diagnosis

- You can use CHA to diagnose and triage past problems.
- You specify the past dates through EMCC Incident Manager or through the command-line interface CHACTL.

```
[grid@host03 ~]$ chactl query diagnosis -cluster

2017-04-03 15:22:21.0  Host host03  Host Memory Consumption [detected]
2017-04-03 15:22:33.0  Host host04  Host Memory Consumption [detected]

Problem: Host Memory Consumption
Description: CHA detected that more memory than expected is consumed on this server. The memory is
not allocated by sessions of this database.
Cause: The Cluster Health Advisor (CHA) detected an increase in memory consumption by other
databases or by applications not connected to a database on this node.
Action: Identify the top memory consumers by using the Cluster Health Monitor (CHM).
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Cluster Health Advisor stores the analysis results, along with diagnosis information, corrective action, and metric evidence for later triage in GIMR. It also sends warning messages to EMCC by using the Oracle Clusterware event notification protocol.

You can also use Oracle Cluster Health Advisor to diagnose and triage past problems. You specify the past dates through Oracle Enterprise Manager Cloud Control (EMCC) Incident Manager or through the command-line interface CHACTL. The capability of Oracle Cluster Health Advisor to review past problems is managed by the retention setting for Oracle Cluster Health Advisor's tablespace in GIMR. The default retention period is 72 hours.

Use the chactl query model command to view the model details. For example:

```
$ chactl query model -name weekday
Model: weekday
Target Type: CLUSTERWARE
Version: OS12.2_V14_0.9.8
OS Calibrated on: Linux amd64
Calibration Target Name: CLUSTER01
Calibration Date: 2017-04-05 01:13:49
Calibration Time Ranges: start=2017-04-03 20:50:00,end=2017-04-04 15:00:00
Calibration KPIs: not specified
```

Managing the CHA Repository

- Use `chactl query repository` to view the maximum retention time, number of targets, and the CHA repository size.

```
$ chactl query repository
specified max retention time(hrs): 72
available retention time(hrs)      : 1236
available number of entities       : 17
allocated number of entities       : 0
total repository size(gb)         : 15.00
allocated repository size(gb)     : 0.07
```

- To set the maximum retention time, use the following:

```
$ chactl set maxretention -time 96
max retention successfully set to 96 hours
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Use the `chactl query repository` command to view the maximum retention time, number of targets, and size of the Cluster Health Advisor repository.

Use the `chactl set maxretention` command to set the maximum retention time for diagnostic data. The default and minimum retention time is 72 hours. If the Cluster Health Advisor repository does not have enough space, the retention time is decreased for all the targets.

To increase the size of the Cluster Health Advisor repository, use the `chactl resize repository` command. For example, to resize the repository to support 32 targets by using the currently set maximum retention time, you would use the following command:

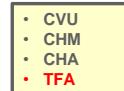
```
$ chactl resize repository -entities 32
repository successfully resized for 32 targets
```

- MDBFG (GIMR foregrounds)
- ASMFG (ASM foregrounds)
- CLUST (Cluster)
- OTHER (other processes)

```
# oclumon dumpnodeview -procag
dumpnodeview: Node name not given. Querying for the local host
-----
Node: host01 Clock: '2017-03-27 20.34.10+0000' SerialNo:89599
-----
PROCESS AGGREGATE:

cpuusage[%]    privatemem[KB]    maxshmem[KB]    #threads    #fd    #proc    category    sid
      18.19        10082000        2846344        58        2474        54   MDBBG    -MGMTDB
       1.65        4171836         2796268         9        151         9   MDBFG    -MGMTDB
      19.02        1018892          74640        33        2001        31   ASMBG    +ASM1
       0.00        519860           55484        14        710        14   ASMFG    +ASM1
      50.45        2664620          117864        498        4098        31    CLUST
      10.66        472792          21124        182        1682        170    OTHER
```

Trace File Analyzer (TFA) Collector



- TFA Collector is a diagnostic collection utility that simplifies diagnostic data collection for:
 - Oracle Grid Infrastructure
 - Oracle RAC systems
- TFA is automatically configured as part of the Oracle Grid Infrastructure configuration when running `root.sh` or `root upgrade.sh`
- TFA can automatically collect diagnostic information when it detects that an incident (like ORA-600, ORA-7445, ORA-4031, node evictions, and instance termination) has occurred.
- With the release of Oracle Database 12.2, Oracle Trace File Analyzer is built on Java Runtime Environment (JRE) 1.8.

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The Oracle Trace File Analyzer (TFA) Collector is a diagnostic collection utility that simplifies diagnostic data collection for Oracle Grid Infrastructure and Oracle RAC systems. Oracle Trace File Analyzer is automatically configured as part of the Oracle Grid Infrastructure configuration when running `root.sh` or `rootupgrade.sh`. Two `tfa` directories are created when Oracle Trace File Analyzer is installed as part of the Oracle Grid Infrastructure.

- **`Grid_home/tfa`:** This directory contains the Oracle Trace File Analyzer executables and some configuration files.
- **`ORACLE_BASE/tfa`:** Where `ORACLE_BASE` is the Oracle Grid Infrastructure owner's `ORACLE_BASE`. This directory contains the Oracle Trace File Analyzer metadata files and logs.

TFA can automatically collect diagnostic information when it detects that an incident has occurred. TFA also has a web-based visualization feature that adds easy-to-navigate, web-based visualization to TFA that is installed as part of Oracle Grid Infrastructure. You can use TFA to efficiently review and analyze diagnostic information that is gathered as part of a TFA collection.

Using TFA, you can choose between automated and manual collection of diagnostic data, which can then either be analyzed directly or as a data stream to auxiliary systems, such as My Oracle Support, to be visualized or analyzed in a certain context.

TFA Collector Utility

```

# /u01/app/12.2.0/grid/tfa/bin/tfactl -h
start          Starts TFA
stop           Stops TFA
enable          Enables TFA Auto restart
disable         Disables TFA Auto restart
print           Prints requested details
access          Adds, Removes, or Lists TFA Users and Groups
purge           Deletes collections from TFA repository
directory       Adds, Removes, or Modifies directory in TFA
host            Adds or Removes host in TFA
receiver        Adds or Removes receiver in TFA
diagcollect     Collects logs from across nodes in cluster
collection      Manages TFA Collections
analyze         Lists events summary and search strings in alert logs
set             Turns ON/OFF or Modifies various TFA features
ips             Executes IPS (Incident Package Service) commands
...
uninstall       Uninstalls TFA from this node
diagnosetfa    Collects TFA Diagnostics
For help with a command: /u01/app/12.2.0/grid/tfa/bin/tfactl
<command> -help
  
```

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The `/u01/app/12.2.0/grid/tfa/bin/tfactl -h` utility is used to complete collection operations and other TFA Collector configuration operations. Some of these operations can be completed by `root` only and others by DBAs.

The operations completed by `root` only are `start`, `stop`, `enable`, `disable`, `access`, `purge`, `host`, and `set`. The operations that DBAs can complete are `diagcollect`, `analyze`, `directory` and `print`.

There are other commands listed in the slide. You can observe that the `IPS` command is fully integrated into TFA Collector as it is with ADR Command Interpreter `adrci`.

To get detailed information about the usage of each command, use `help <command>`.

```

# /u01/app/12.2.0/grid/tfa/bin/tfactl
tfactl> help diagcollect
Usage: /u01/app/oracle/tfa/bin/tfactl diagcollect [-all | -database
<all|d1,d2..> | -asm | -dbwlm | -tns | -crs | -wls | -emagent | -oms | -ocm | -
emplugins | -em | -acfs | -install | -cfgtools | -os | -ips [-oraclehome
<oracle_home>] [-adrbasepath <adr_basepath>] [-adrhomepath <adr_homepath>] [-
level <corr_level>] [-incident <adr_incident> | -problem <prob_id> | -
problemkey <prob_key>] ...
  
```

TFA Collector Analysis



| Which Files | How |
|-----------------------------|-----------------------------|
| CRS, ASM, and DB alert logs | By component |
| System logs | By time |
| | By specific search patterns |
| | By type (error, warning) |

- Show summary of events from alert logs, system messages in last 5 hours.

```
# tfactl analyze -since 5h
```

- Show summary of events from system messages in last one day.

```
# tfactl analyze -comp os -since 1d
```

- Search string ORA- in alert and system logs in the past two days.

```
# tfactl analyze -search "ORA-" -since 2d
```

- Run oratop in batch mode for database cdb1.

```
# tfactl analyze -comp oratop -database cdb1 -bn1
```

ORACLE®

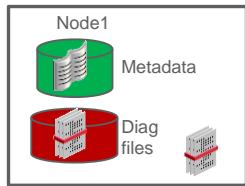
Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The ANALYZE command of the tfactl utility allows you to quickly get diagnostic information about events and also search strings in diagnostic files.

TFA Collector Repository

Check storage used by TFA Collector on nodes repository.

10gb
by default



```
# tfactl print config
```



- 30Gb in the repository



No more collections when maximum configured size of repository reached



```
# tfactl set reposizeMB=100000
```

```
# tfactl purge -older 30d
```

```
# tfactl set repositorydir=newreposdir
```

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

It is important to check when the repository for collections is filled up. The `print` command of the `tfactl` utility displays the current space used by the repository for all collections. TFA Collector does not generate any new collection when the maximum size of the repository is reached, avoiding the filling of file systems by new collections.

You can either increase the maximum size for the repository or delete existing collections (older than 30 days in the example in the slide) or even change the location of the repository to another directory on another file system.

Managing ADR Logs by Using tfactl managelogs

- To limit the purge or show operations only to files that are older than a specified time, use:

```
$ tfactl managelogs -older nm|h|d Files from past 'n' [d]ays or 'n' [h]ours or 'n' [m]inutes
```

- Use the `-dryrun` option to get an estimate of how many files are removed and how much space is freed.

```
$ tfactl managelogs -purge -older 30d -dryrun
Output from host : host01
-----
2017-04-06 17:14:46: INFO Estimating files older than 30 days

2017-04-06 17:14:46: INFO Estimating purge for diagnostic destination
  "diag/asm/user_root/host_969082260_107" for files ~ 1 files deleted , 9.66 KB freed ]
2017-04-06 17:14:46: INFO Estimating purge for diagnostic destination "diag/asm/+asm/+ASM1" for
  files ~ 572 files deleted , 1.74 MB freed ]
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Use the `managelogs` command to manage the Automatic Diagnostic Repository log and trace files. The `-purge` command removes the files managed by the Automatic Diagnostic Repository (files are cleared from “ALERT,” “INCIDENT,” “TRACE,” “CDUMP,” “HM,” “UTSCDMP,” and “LOG” under diagnostic destinations) and provides details about the change in the file system space.

For diagnostic destinations where large numbers of files exist, the command might take a while. Check the removal in progress from corresponding directories. You must have operating system privilege over the corresponding diagnostic destinations to remove the files.

To limit the purge or show operations only to files that are older than a specified time, use the `-older` option with `-purge`:

```
$ tfactl managelogs -purge -older 30d
```

Perform a dry run to get an estimate of how many files are removed and how much space is freed by executing the `purge` command with the `-dryrun` option.

Lesson Agenda

- Using Oracle Autonomous Health Framework
 - Cluster Verify Utility (CVU)
 - Cluster Health Monitor (CHM)
 - Cluster Health Advisor (CHA)
 - Trace File Analyzer (TFA) Collector
- **Using the Cluster Resource Activity Log (CALOG)**
- Using Oracle Clusterware Diagnostic and Alert Log Data
- Node Eviction



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Cluster Resource Activity Log (CALOG)

- When a Clusterware-managed resource fails, Clusterware alerts you by logging messages about the failure in GIMR.
- The cluster resource activity log increases the amount of information available for a particular resource failure.
- Every write to a log file is tagged with an activity ID, which can be nested.

```
----ACTIVITY START----  
timestamp : 2017-05-08 14:12:31.127-08:00  
writer_process_id : 7938  
writer_process_name : ~/oracle/bin/crsd
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

If an Oracle Clusterware-managed resource fails, then Oracle Clusterware logs messages about the failure in the cluster resource activity log located in the Grid Infrastructure Management Repository. The cluster resource activity log increases the amount of information that is available to you for a particular resource failure, by providing you with a unified view of why a resource failed. This is advantageous when you have multiple resources with complicated dependencies between them. The cluster resource activity log is an adjunct to current Oracle Clusterware logging and alert log messages.

One feature of the cluster resource activity log is that every write to a log file is tagged with an activity ID, which can be nested. For example, if you have a stack running and you run the `crsctl stop clusterware -all` command, all activities written while this command is running are tagged with the same parent activity ID. On each node, the command creates sub-IDs under the parent IDs, and tags each respective activity with its own activity ID. Further, each resource on each of the individual nodes creates its own sub-ID based on its parent ID, thus creating a hierarchy of activity IDs. This enables you to analyze the data to find specific activities.

For example, you may have many resources with complicated dependencies among one another, and with a database service. On Friday, you see that all the resources are running on one node but when you return on Monday, every resource is on a different node, and you want to know why.

You can query the cluster resource activity log for all activities involving these resources and the database service, and see a complete flow (which, if you have thousands of nodes and thousands of resources, may consist of gigabytes of data, layered not by node or resource, but by time). You could also query each sub-ID within the parent service failover ID, and see what happened and why, specifically.

Note: Oracle Clusterware does not write messages that contain security-related information, for example, logon credentials, to the cluster activity log.

Querying and Managing the CALOG

- Use the `crsctl query calog` command to monitor Clusterware-managed resource activity.

```
$ crsctl query calog
2017-03-01 22:11:56.813000 : Starting Oracle Cluster Ready Services managed resources on
    server 'host01' : 148840631681110149/0/1 :
...
```

- Use `crsctl get calog maxsize` to query the maximum space allotted to cluster resource activity information.

```
$ crsctl get calog maxsize
CRS-6760: The maximum size of the cluster activity log is 512 MB.
```

- Use `crsctl set calog maxsize` to set the maximum space allotted to cluster resource activity information.

```
$ crsctl set calog maxsize 1024
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

When an Oracle Clusterware–managed resource fails due to a problem with the resource itself, with the hosting node, or with the network, Oracle Clusterware logs messages about the failure in a text file.

The cluster resource activity log provides precise and specific information about a resource failure, separate from diagnostic logs. It also provides a unified view of the cause of resource failure. Use the following commands to manage and view the contents of the Clusterware resource activity log:

- To query the cluster resource activity logs that match specific criteria, use: `crsctl query calog`.
- To store Oracle Clusterware–managed resource activity information, query the maximum space allotted to the cluster resource activity log: `crsctl get calog maxsize`.
- Query the retention time of the cluster activity log by using: `crsctl get calog retentiontime`.
- Configure the maximum amount of space allotted to store Oracle Clusterware–managed resource activity information: `crsctl set calog maxsize`.
- Configure the retention time of the cluster resource activity log: `crsctl set calog retentiontime`.

Lesson Agenda

- Using Oracle Autonomous Health Framework
- Using the Cluster Resource Activity Log (CALOG)
- Using Oracle Clusterware Diagnostic and Alert Log Data
 - ADR Directory Structure
 - Files in the Trace Directory
 - Clusterware Trace Files
 - The Oracle Clusterware Alert Log
 - Incident Trace Files
 - Other Diagnostic Data
- Node Eviction



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

ADR Directory Structure

- Clusterware uses Oracle Database fault diagnosability infrastructure to manage diagnostic data and its alert log.
 - As a result, most diagnostic data resides in the Automatic Diagnostic Repository (ADR).
- Clusterware ADR data is written under a root directory known as the ADR base.
- Components other than ADR use this directory, so it may also be referred to as the Oracle base.
- The ADR home for Clusterware on a given host has the structure:
ORACLE_BASE/diag/crs/<host_name>/crs
- The ADR/Oracle base location can only be changed if you reinstall the Oracle Grid Infrastructure.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware uses Oracle Database fault diagnosability infrastructure to manage diagnostic data and its alert log. As a result, most diagnostic data resides in the Automatic Diagnostic Repository (ADR), a collection of directories and files located under a base directory that you specify during installation.

Clusterware ADR data is written under a root directory known as the ADR base. Because components other than ADR use this directory, it may also be referred to as the Oracle base. You specify the file system path to use as the base during Oracle Grid Infrastructure installation and can only be changed if you reinstall the Oracle Grid Infrastructure.

ADR files reside in an ADR home directory. The ADR home for Oracle Clusterware running on a given host always has this structure: ORACLE_BASE/diag/crs/<host_name>/crs

In the preceding example, ORACLE_BASE is the Oracle base path you specified when you installed the Oracle Grid Infrastructure and <host_name> is the name of the host. On a Windows platform, this path uses backslashes (\) to separate directory names.

Under the ADR home are various directories for specific types of ADR data. The directories of greatest interest are trace and incident. The trace directory contains all normal (non-incident) trace files written by Oracle Clusterware daemons and command-line programs as well as the simple text version of the Oracle Clusterware alert log. This organization differs significantly from versions before Oracle Clusterware 12c release 1 (12.1.0.2), where diagnostic log files were written under distinct directories per daemon.

Files in the Trace Directory

- Starting with Clusterware 12.1.0.2, diagnostic data files written by Clusterware programs are known as trace files.
 - The trace files have a `.trc` file extension.
 - Trace files appear in the `trace` subdirectory of the ADR home.

```
$ ls -l $ORACLE_BASE/diag/crs/host01/crs
drwxrwxr-x  2 root oinstall 4096 Jun 16 14:55 alert
drwxrwxr-x 10 root oinstall 4096 Jun 18 06:12 cdump
drwxrwxr-x 10 root oinstall 4096 Jun 18 06:12 incident
drwxrwxr-x  2 root oinstall 4096 Jun 16 14:54 incpkg
drwxrwxr-x  2 root oinstall 4096 Jun 18 06:12 lck
drwxrwxr-x  4 root oinstall 4096 Jun 16 14:54 log
drwxrwxr-x  2 root oinstall 4096 Jun 16 14:54 metadata
drwxrwxr-x  2 root oinstall 4096 Jun 16 14:54 metadata_dgif
drwxrwxr-x  2 root oinstall 4096 Jun 16 14:54 metadata_pv
drwxrwxr-x  2 root oinstall 4096 Jun 18 06:12 stage
drwxrwxr-x  2 root oinstall 4096 Jun 18 06:12 sweep
drwxrwxr-x  2 grid oinstall 77824 Jun 21 07:56 trace
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Starting with Oracle Clusterware 12.1.0.2, diagnostic data files written by Oracle Clusterware programs are known as trace files and have a `.trc` file extension, and appear together in the trace subdirectory of the ADR home. The naming convention for these files generally uses the executable program name as the file name, possibly augmented with other data depending on the type of program. Trace files written by Clusterware command-line programs incorporate the O/S process ID (PID) in the trace file name to distinguish data from multiple invocations of the same command program. For example, trace data written by `crsctl` uses this name structure: `crsctl_PID.trc`. In this example, PID is the operating system process ID displayed as decimal digits:

```
-rw-rw---- 1 grid    oinstall 1005 Jun 19 07:53 crsctl_10173.trc
-rw-rw---- 1 grid    oinstall 1996 Jun 19 19:48 crsctl_10230.trc
```

Trace files written by Clusterware daemon programs do not include a PID in the file name, and they also are subject to a file rotation mechanism that affects naming. Rotation means that when the current daemon trace file reaches a certain size, the file is closed, renamed, and a new trace file is opened. This occurs a fixed number of times, and then the oldest trace file from the daemon is discarded, keeping the rotation set at a fixed size.

Clusterware Trace Files

- Most Clusterware daemons use a size limit of 10 MB and a rotation of 10 files, for a total of 100 MB of trace data.
- The current trace file for a given daemon simply uses the program name as the file name.
 - Older files in the rotation append a number to the file name.

```
-rw-rw---- 1 root oinstall 10488172 Jun 18 05:01 ohasd_1.trc
-rw-rw---- 1 root oinstall 10486338 Jun 19 04:51 ohasd_2.trc
-rw-rw---- 1 root oinstall 7161272 Jun 21 09:45 ohasd.trc
```

- Clusterware agent trace files are subject to special naming conventions indicating the origin of the agent.

```
-rw-rw---- 1 oracle oinstall 635252 Jun 21 09:27 crsd_oraagent_oracle.trc
-rw-rw---- 1 root oinstall 106538 Jun 21 09:27 crsd_orarootagent_root.trc
-rw-rw---- 1 grid oinstall 7101665 Jun 21 09:27 crsd_scriptagent_grid.trc
-rw-rw---- 1 grid oinstall 466404 Jun 21 09:45 ohasd_oraagent_grid.trc
-rw-rw---- 1 root oinstall 1048622 Jun 17 3:14 ohasd_orarootagent_root_1.trc
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Most Clusterware daemons use a file size limit of 10 MB and a rotation of 10 files, thus maintaining a total of 100 MB of trace data. The current trace file for a given daemon simply uses the program name as the file name; older files in the rotation append a number to the file name. For example, the trace file currently being written by the Oracle High Availability Services daemon is named `ohasd.trc`; the most recently rotated-out trace file is named `ohasd_n.trc`, where *n* is an ever-increasing decimal integer. The file with the highest *n* is actually the most recently archived trace, and the file with the lowest *n* is the oldest.

Clusterware agents are daemon processes whose trace files are subject to special naming conventions indicating the origin of the agent (whether it was spawned by the OHASD or the Cluster Ready Services daemon [CRSD]) and the O/S username with which the agent runs. Thus, the name structure for agents is: `origin_executable_<user_name>`

In the previous example, `origin` is either `ohasd` or `crsd`, `executable` is the executable program name, and `<user_name>` is the O/S username. In addition, because they are daemons, agent trace files are subject to the rotation mechanism previously described, so files with an additional `_n` suffix are present after rotation occurs as shown in the slide.

```
-rw-rw---- 1 oracle oinstall 6352152 Jun 21 09:27 crsd_oraagent_oracle.trc
-rw-rw---- 1 root oinstall 1065380 Jun 21 09:27 crsd_orarootagent_root.trc
-rw-rw---- 1 grid oinstall 7101665 Jun 21 09:27 crsd_scriptagent_grid.trc
-rw-rw---- 1 grid oinstall 466404 Jun 21 09:45 ohasd_oraagent_grid.trc
-rw-rw---- 1 root oinstall 1048622 Jun 17 3:14 ohasd_orarootagent_root_1.trc
```

The Oracle Clusterware Alert Log

- The `trace` subdirectory in the Clusterware ADR home contains the simple text Oracle Clusterware alert log.
- The alert log is also written as an XML file in the `alert` subdirectory of the ADR home, but the text alert log is most easily read.

```
$ ls -l /u01/app/grid/diag/crs/host01/crs/trace/*.log  
-rw-rw---- 1 root oinstall 208779 Jun 21 07:57 alert.log  
  
$ ls -l /u01/app/grid/diag/crs/host01/crs/alert  
-rw-rw---- 1 root oinstall 501054 Jun 21 07:57 log.xml
```

- Unlike the Database alert log, Clusterware alert log messages are identified, documented, and translated.
- Messages and data written to Oracle Clusterware trace files generally are not documented and translated.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Besides trace files, the trace subdirectory in the Oracle Clusterware ADR home contains the simple text Oracle Clusterware alert log. It always has the name `alert.log`. The alert log is also written as an XML file in the `alert` subdirectory of the ADR home, but the text alert log is most easily read.

The alert log is the first place to look when a problem or issue arises with Oracle Clusterware. Unlike the Oracle Database instance alert log, messages in the Oracle Clusterware alert log are identified, documented, and localized (translated). Oracle Clusterware alert messages are written for most significant events or errors that occur.

2015-06-18 07:44:33.426 [OCSSD(12341)]CRS-1601: CSSD Reconfiguration complete.
Active nodes are host01 .

2015-06-18 07:44:35.457 [OCTSSD(13166)]CRS-8500: Oracle Clusterware OCTSSD process is starting with operating system process ID 13166

2015-06-18 07:44:36.569 [OCTSSD(13166)]CRS-2401: The Cluster Time Synchronization Service started on host host01.

2015-06-18 07:44:36.570 [OCTSSD(13166)]CRS-2407: The new Cluster Time Synchronization Service reference node is host host01.

Incident Trace Files

- Certain errors occur in Oracle Clusterware programs that will raise an ADR incident.
- In most cases, these errors should be reported to My Oracle Support for diagnosis.
- In addition to alert messages, incidents also cause the affected program to produce a separate trace file containing diagnostic data related to the error.
- These incident-specific trace files are collected in the `incident` subdirectory of the ADR home.

```
$ ls -l /u01/app/grid/diag/crs/host01/crs/incident/incdir_1
-rw-rw---- 1 grid oinstall 3080542 Jun 17 18:23 ocssd_il1.trc
-rw-rw---- 1 grid oinstall 526885 Jun 17 18:23 ocssd_il1.trm
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Certain errors occur in Oracle Clusterware programs that will raise an ADR incident. In most cases, these errors should be reported to My Oracle Support for diagnosis. The occurrence of an incident normally produces one or more descriptive messages in the Oracle Clusterware alert log.

In addition to alert messages, incidents also cause the affected program to produce a special, separate trace file containing diagnostic data related to the error. These incident-specific trace files are collected in the `incident` subdirectory of the ADR home rather than the `trace` subdirectory. Both the normal trace files and incident trace files are collected and submitted to Oracle when reporting the error.

Other Diagnostic Data

- Besides ADR data, Oracle Clusterware collects or uses other data related to problem diagnosis.
- This data resides under the same ADR base but a separate directory structure:
ORACLE_BASE/crsdata/<host_name>

```
$ ls -l $ORACLE_BASE/crsdata/host01
drwxrwxr-x 13 grid oinstall 4096 Jun 16 14:59 .
drwxrwxr-x  4 grid oinstall 4096 Jun 16 14:54 ..
drwxr-xr-x  2 root root    4096 Jun 16 15:00 acfs
drwxrwxrwt  2 grid oinstall 4096 Jun 17 12:13 core
drwxrwxr-x  2 grid oinstall 4096 Jun 16 14:55 crsconfig
drwxrwxr-x  2 grid oinstall 4096 Jun 16 14:54 crsdiag
drwxrwxr-x  4 grid oinstall 4096 Jun 16 23:06 cvu
drwxr-x--T  2 grid oinstall 4096 Jun 21 01:15 evm
drwxrwxrwt  2 grid oinstall 4096 Jun 16 23:42 output
drwxr-xr-x  2 grid oinstall 4096 Jun 16 14:54 ovmmwallet
drwxrwxr-x  2 grid oinstall 4096 Jun 16 14:54 rhp
drwxr-x---  2 grid oinstall 4096 Jun 16 14:54 scripts
drwxr-xr-x  3 grid oinstall 4096 Jun 16 14:54 trace
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Besides ADR data, Oracle Clusterware collects or uses other data related to problem diagnosis. Starting with Oracle Clusterware 12c release 1 (12.1.0.2), this data resides under the same base path used by ADR, but in a separate directory structure with this form: ORACLE_BASE/crsdata/<host_name>. In this example, ORACLE_BASE is the Oracle base path you specified when you installed the Grid Infrastructure and <host_name> is the name of the host.

In this directory, on a given host, are several subdirectories. The two subdirectories of greatest interest if a problem occurs are named `core` and `output`. The `core` directory is where Oracle Clusterware daemon core files are written when the normal ADR location used for core files is not available (for example, before ADR services are initialized in a program). The `output` directory is where Oracle Clusterware daemons redirect their C standard output and standard error files. These files generally use a name structure consisting of the executable name with the characters OUT appended to a `.trc` file extension (like trace files). For example, the redirected standard output from the Cluster Time Synchronization Service daemon is named `octssdOUT.trc`.

```
$ ls -l $ORACLE_BASE/crsdata/host01/output/octssdOUT.trc
-rw-r--r-- 1 root root 3150 Jun 18 07:44 octssdOUT.trc
```

Typically, daemons write very little to these files, but in certain failure scenarios important data may be written there.

Lesson Agenda

- Using Oracle Autonomous Health Framework
- Using the Cluster Resource Activity Log (CLOG)
- Using Oracle Clusterware Diagnostic and Alert Log Data
 - ADR Directory Structure
 - Files in the Trace Directory
 - Clusterware Trace Files
 - The Oracle Clusterware Alert Log
 - Incident Trace Files
 - Other Diagnostic Data
- Node Eviction



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Node Eviction: Overview

- Clusterware will evict one or more nodes from the cluster if a critical problem is detected. These problems include:
 - A node not responding via a network or disk heartbeat
 - A hung or severely degraded node
 - A hung `ocssd.bin` process
- In a Rebootless Node Eviction (restart), an eviction may cause a graceful restart of the Clusterware stack.
- Node Eviction (reboot) may happen if
 - The check for a successful kill of the IO processes fails
 - CSSD gets killed during the operation
 - `cssdmonitor` is not scheduled
 - The Clusterware stack cannot be shutdown



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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.

Starting with 11.2.0.2, a node eviction may not actually reboot the machine. This is called a *rebootless restart*. In this case we restart most of the Clusterware stack to see if that fixes the unhealthy node.

Rebootless Node Eviction: Example

- Clusterware alert.log file

```
2017-11-17 12:15:12.758 [OCSSD(18455)]CRS-1612: Network communication with node host03 (2) missing for 50% of timeout interval. Removal of this node from cluster in 14.520 seconds ...
2017-11-17 12:15:28.320 [OCSSD(18455)]CRS-1608: This node was evicted by node 2, host03; details at (:CSSNM00005:) in /u01/app/grid/diag/crs/host01/crs/trace/ocssd.trc.
2017-11-17 12:15:28.341 [OCSSD(18455)]CRS-1652: Starting clean up of CRSD resources.
2017-11-17 12:15:34.939 [OCSSD(18455)]CRS-1654: Clean up of CRSD resources finished successfully.
```

- Check OHASD service status

```
[root@host01 ~]# crsctl check crs
CRS-4638: Oracle High Availability Services is online
CRS-4535: Cannot communicate with Cluster Ready Services
CRS-4530: Communications failure contacting Cluster Synchronization Services daemon
CRS-4534: Cannot communicate with Event Manager
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The example in the slide is the Rebootless node eviction. Instead of fast rebooting the node, a graceful shutdown of the stack is attempted. The OHASD service is still up and running to restart the clusterware stack when the problem is resolved.

Even though Node Eviction (reboot) will be seen less, but it may still happen if

- The check for a successful kill of the IO processes fails
- CSSD gets killed during the operation
- cssdmonitor is not scheduled
- The CW stack cannot be shutdown

Processes Roles For Node Reboots

- `ocssd`
 - Internode health monitoring via Network & Disk Heartbeat
- `cssdagent` and `cssdmonitor`
 - Monitoring for node hangs
 - Monitoring to the OCSSD process for hangs
 - Monitoring Vendor clusterware
- `oclskd`
 - Handles RDBMS Instance Hang-up issues causing Instance Eviction
 - Is used by CSS to reboot a node based on requests from other nodes in the cluster (Member Kill Escalation)



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The `ocssd` process is responsible for internode 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 multithreaded process that runs at an elevated priority and runs as the `Oracle` user.

The `cssdagent` process is spawned by `ohasd` and is responsible for spawning the `ocssd` process, monitoring node hangs, and monitoring the `ocssd` process for hangs, and monitoring vendor clusterware (via vmon functionality). This is a multithreaded 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 nonvendor clusterware environments.

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.

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

Reboot Advisory in clusterware alert.log

- Clusterware may instigate rebooting of a node in certain circumstances to ensure the overall health of the cluster.
- When the decision is made to reboot the problematic node, ordinary activity logging is not reliable.
 - Time is of the essence in most reboot scenarios.
 - The reboot usually occurs before the operating system flushes buffered log data to disk.
- This means that an explanation of what led to the reboot may be lost.
- The Reboot Advisory feature addresses this in two ways:
 - The reboot decision is written to a small file, normally on local storage using a direct, nonbuffered I/O request.
 - The reboot decision is broadcast over all available network interfaces on the failed node.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Clusterware may, in certain circumstances, instigate rebooting of a node to ensure the overall health of the cluster and of the databases and other applications running on it. The decision to reboot a node can be made by Clusterware running on that node or by Clusterware on another cluster node. When the decision is made on the problematic node, ordinary activity logging (such as the Clusterware alert log) is not reliable: time is of essence in most reboot scenarios, and the reboot usually occurs before the operating system flushes buffered log data to disk. This means that an explanation of what led to the reboot may be lost.

There is a feature called Reboot Advisory that improves the odds of preserving an explanation for a Clusterware-initiated reboot. At the moment a reboot decision is made by Clusterware, a short explanatory message is produced and an attempt is made to “publish” it in two ways:

- The reboot decision is written to a small file, normally on local storage using a direct, nonbuffered I/O request. The file is created and preformatted in advance of the failure (during Clusterware startup), so this I/O has a high probability of success, even on a failing system.
- The reboot decision is broadcast over all available network interfaces on the failed node.

These operations are executed in parallel and are subject to an elapsed time limit so as not to delay the impending reboot.

Reboot Advisory in clusterware alert.log

- Attempting both disk and network publication of the message makes it likely that at least one succeeds.
- Successfully stored or transmitted Reboot Advisories ultimately appear in a Clusterware alert log on one or more cluster nodes.
- Reboot Advisories use the same alert log messages, normally two per advisory.
 - The first is message is a CRS-8011:

```
[ohasd(24687)]CRS-8011:reboot advisory message from host: host01, component: CSSMON, with timestamp: L-2013-06-04-02:09:31.220
```

- The second is a CRS-8013, which conveys the explanatory message for the forced reboot:

```
[ohasd(24687)]CRS-8013:reboot advisory message text: Rebooting after limit 28500 exceeded; disk timeout 27630, network timeout 28500, last heartbeat from ocssd at epoch seconds 1241543005.340, 4294967295 milliseconds ago based on invariant clock value of 93235653
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

When network broadcast of a Reboot Advisory is successful, the associated messages appear in the alert logs of other nodes in the cluster. This happens more or less instantaneously, so the messages can be viewed immediately to determine the cause of the reboot. The message includes the host name of the rebooted node to distinguish it from the normal flow of alert messages for that node. Only nodes in the same cluster as the failing node will display these messages.

If the Reboot Advisory was successfully written to a disk file, when Oracle Clusterware starts the next time on that node, it will produce messages related to the prior in the Clusterware alert log. Reboot Advisories are timestamped and the startup scan for these files will announce any occurrences that are less than three days old. The scan does not empty or mark already-announced files, so the same Reboot Advisory can appear in the alert log multiple times if Clusterware is restarted on a node multiple times within a three-day period.

Whether from a file or a network broadcast, Reboot Advisories use the same alert log messages, normally two per advisory. The first is message CRS-8011, which displays the host name of the rebooting node, a software component identifier, and a time stamp (approximately the time of the reboot). An example looks like this:

```
[ohasd(24687)]CRS-8011:reboot advisory message from host: host01, component: CSSMON, with timestamp: L-2009-05-05-10:03:25.340
```

Following message CRS-8011 will be CRS-8013, which conveys the explanatory message for the forced reboot, as in this example:

```
[ohasd(24687)]CRS-8013:reboot advisory message text: Rebooting after limit  
28500 exceeded; disk timeout 27630, network timeout 28500, last heartbeat from  
ocssd at epoch seconds 1241543005.340, 4294967295 milliseconds ago based on  
invariant clock value of 93235653
```

Note that everything in message CRS-8013 after “text:” originates in the Clusterware component that instigated the reboot. Because of the critical circumstances in which it is produced, this text does not come from an Oracle NLS message file: it is always in English language and USASCII7 character set.

In some circumstances, Reboot Advisories may convey binary diagnostic data in addition to a text message. If so, message CRS-8014 and one or more of message CRS-8015 will also appear. This binary data is used only if the reboot situation is reported to Oracle for resolution.

Because multiple components can write to the Clusterware alert log at the same time, it is possible that the messages associated with a given Reboot Advisory may appear with other (unrelated) messages interspersed. However, messages for different Reboot Advisories are never interleaved: all of the messages for one Advisory are written before any message for another Advisory.

Here are the common reasons of Node Evictions.

COMMON CAUSES OF OCSSD EVICTIONS

- Network failure or latency between nodes. It would take 30 consecutive missed checkins (by default - determined by the CSS misscount) to cause a node eviction.
- Problems writing to or reading from the CSS voting disk. If the node cannot perform a disk heartbeat to the majority of its voting files, then the node will be evicted.
- A member kill escalation. For example, database LMON process may request CSS to remove an instance from the cluster via the instance eviction mechanism. If this times out it could escalate to a node kill.
- An unexpected failure or hang of the OCSSD process, this can be caused by any of the above issues or something else.
- An Oracle bug.

COMMON CAUSES OF CSSDAGENT OR CSSDMONITOR EVICTIONS

- An OS scheduler problem. For example, if the OS is getting locked up in a driver or hardware or there is excessive amounts of load on the machine (at or near 100% cpu utilization), thus preventing the scheduler from behaving reasonably.
- A thread(s) within the CSS daemon hung.
- An Oracle bug.

Other Log & Trace Files to Review

- CHM data or OS Watcher
- ocssd log(s):
 - \$ORACLE_BASE/diag/crs/<host>/crs/trace/ocssd.trc
- cssdagent log(s):
 - \$ORACLE_BASE/diag/crs/<host>/crs/trace/ohasd_cssdagent_root.trc
- cssdmonitor log(s):
 - \$ORACLE_BASE/diag/crs/<host>/crs/trace/ohasd_cssdmonitor_root.trc
- Messages file (Linux):
 - /var/log/messages



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

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 `$ORACLE_BASE/diag/crs/<hostname>/crs/trace/ocssd.log` may also contain messages similar to "Begin Dump" or "End Dump" just before the reboot.

Other useful log files include the Clusterware alert log in `$ORACLE_BASE/diag/crs/<hostname>/crs/trace` 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.

- Example message from a clusterware alert log:

```
[ohasd(11243)]CRS-8011:reboot advisory message from host: sta00129,  
component: cssagent, with timestamp: L-Date-Time  
[ohasd(11243)]CRS-8013:reboot advisory message text: Rebooting after limit  
28500 exceeded; disk timeout 27630, network timeout 28500, last heartbeat from  
CSSD at epoch seconds 1241543005.340, 4294967295 milliseconds ago based on  
invariant clock value of 93235653
```

- Example of an OCSSD eviction due to loss of voting disk:

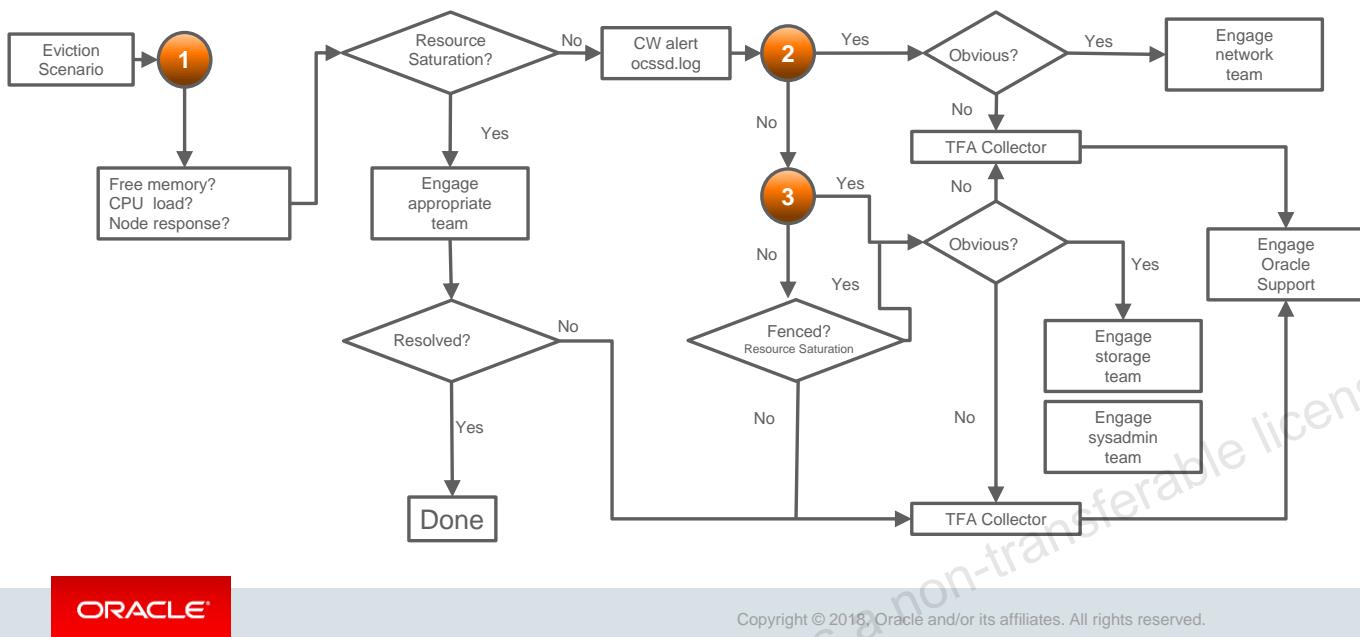
CSS log:

```
<Timestamp>:[ CSSD][1100548416](:CSSNM00018:)clssnmvDiskCheck: Aborting, 0 of 3  
configured voting disks available, need 2  
<Timestamp>:[ CSSD][1100548416]#####  
<Timestamp>:[ CSSD][1100548416]clsscExit: CSSD aborting from thread  
clssnmvDiskPingMonitorThread
```

OS messages:

```
<Timestamp> choldbr132p kernel: Error:Mpx:All paths to Symm 000190104720 vol  
0c71 are dead.  
<Timestamp> choldbr132p kernel: Error:Mpx:Symm 000190104720 vol 0c71 is dead.  
<Timestamp> choldbr132p kernel: Buffer I/O error on device sdbig, logical block  
0  
...
```

Possible Troubleshooting Scenario: Example



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The diagram in the slide shows a possible troubleshooting scenario for node eviction.

1. Review the following support notes to see if the problem is due to the resource contention.
 - **MOS 1531223.1:** OS Watcher User's Guide
 - **MOS 1328466.1:** Cluster Health Monitor (CHM) FAQ
 - System Log
2. Review the following support notes to see if the problem is due to a network issue.
 - **MOS 1050603.1:** Troubleshooting Clusterware Node Evictions (Reboots)
 - **MOS 1534949.1:** Oracle Grid Infrastructure: How to Troubleshoot Missed Network Heartbeat Evictions
 - **MOS 1546004.1:** Oracle Grid Infrastructure: Understanding Split-Brain Node Eviction
3. Review the following support notes to see if the problem is due to a disk IO issue
 - **MOS 1549428.1:** Oracle Grid Infrastructure: How to Troubleshoot Voting Disk Evictions
 - **MOS 1466639.1:** 11.2.0.3 GI Aborts With "CSSD aborting from thread clssnmvDiskPingMonitorThread" if Only One Voting Disk/File is Configured



Quiz

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

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



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.



Quiz

Which of the following statements is true about Cluster Health Advisor (CHA)?

- a. It is an integral component of the Oracle Autonomous Health Framework.
- b. It continuously monitors cluster nodes and RAC databases for performance and availability issues.
- c. CHA models are conservative to prevent false warning notifications.
- d. All of the above
- e. None of the above



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.



Quiz

Which of the following statements are NOT TRUE about Clusterware Activity Log (CALOG)?

- a. It increases the amount of information available about a particular resource failure.
- b. It decreases the amount of information available to you about resource failures.
- c. Clusterware writes entries that contain security-related information, such as logon credentials, to the cluster activity log.
- d. The `crsctl query calog` command is used to monitor Clusterware-managed resource activity.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Describe the functions of the Cluster Verify Utility
- Describe the functions of the Cluster Health Monitor
- Describe the functions of the Cluster Health Advisor
- Describe the functions of the Trace File Analyzer Collector
- Use the Cluster Resource Activity Log
- Locate the Oracle Clusterware log files
- Troubleshoot Node Eviction



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Practice 10: Overview

This practice covers the following topics:

- 10-1: Working with CLUVFY
- 10-2: Working with Cluster Health Monitor
- 10-3: Working with Cluster Health Advisor
- 10-4: Working with Trace File Analyzer
- 10-5: Cluster Resource Activity Log



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Unauthorized reproduction or distribution prohibited. Copyright© 2019, Oracle and/or its affiliates.

GANG LIU (gangl@baylorhealth.edu) has a non-transferable license
to use this Student Guide.

Making Applications Highly Available with Oracle Clusterware

The Oracle logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe the high-availability components of Oracle Clusterware
- Contrast policy-managed and administration-managed databases
- Create an application Virtual IP (VIP)
- Manage application resources
- Describe and implement Clusterware resource groups



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware High Availability (HA)

Oracle Clusterware provides HA services to Real Application Clusters (RAC) databases and other applications.

- Oracle Clusterware monitors all protected applications periodically.
- Based on the defined failover policy, Oracle Clusterware can restart failed applications on the same node or relocate them to another node.
- It can protect Oracle-based as well as non-Oracle-based applications.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware is a portable cluster infrastructure that provides HA services to RAC databases and other applications. Oracle Clusterware makes applications highly available by monitoring the health of the applications, by restarting applications on failure, and by relocating applications to another cluster node when the currently used node fails or when the application can no longer run in the current node. In the case of node failure, certain types of protected applications, such as a RAC database instance, may not be failed over to the surviving nodes.

A cluster is a collection of two or more nodes where the nodes share a common pool of storage used by the Oracle Clusterware system files (the OCR and the voting disk), a common network interconnect, and a common operating system.

Oracle Clusterware monitors all protected applications periodically, and based on the defined failover policy, it can restart them either on the same node or relocate them to another node, or it can decide to not restart them at all.

Oracle Clusterware HA Components

- Several components are used to implement HA with Clusterware:

| Component | Definition |
|---------------------|--|
| Resource | An entity that Oracle Clusterware manages for HA such as an application |
| Agent | A process that contains the agent framework and user code to manage resources |
| Action Script | An action script defines one or more actions to start, stop, check, or clean resources |
| Privileges | Access and usage privileges for a resource allowing it to run as a different user than the Cluster Ready Services (CRS) user |
| Resource Dependency | A relationship among resources or applications that implies operational ordering |
| Application VIP | A VIP that an application has a dependency with |
| OCR | Storage mechanism for resource profiles, policies, and privileges |



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Several components work together in building a Highly Available (HA) framework for applications using Oracle Clusterware. A resource is an entity that Oracle Clusterware manages for HA such as an application. Resources for HA are defined with an application profile that describes attributes and policies for the application. The application profile also identifies the agent or action script, responsible for providing logic to start, stop, and check the status of a resource. The application profile also defines the failure policies for an application. The Oracle Clusterware software runs with `root` or `administrator` rights. Privileges enable Oracle Clusterware to control the components of an application to include allowing the application to run under the context of a different user from that which Cluster Ready Services (CRS) runs under. Resources can have a dependency on other resources for operation. For example, a database resource may depend on a storage resource to be running. An application Virtual IP (VIP) is a VIP that can fail over to other nodes if policies allow it and is one example of a typical application dependency. The application VIP is a resource. All the information about a resource is stored in the Oracle Clusterware OCR configuration file and is available to each node in the cluster.

Clusterware Resource Modeling

- When an application is registered, you define how Clusterware manages it by defining its attributes.
- The registration information includes an action script or action program that Clusterware calls to start, stop, check, and clean up the application.
 - An action script is a shell script that a generic script agent provided by Oracle Clusterware calls.
 - An application-specific agent is usually a C or C++ program that calls Clusterware APIs directly.
- Oracle Clusterware includes two agents:
 - Script agent: `CRS_HOME/bin/scriptagent`
 - Application agent: `CRS_HOME/bin/appagent`



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

When you register an application as a resource in Oracle Clusterware, you define how Oracle Clusterware manages the application by using resource attributes that you assign to the resource. The frequency with which the resource is checked and the number of attempts to restart a resource on the same server after a failure before attempting to start it on another server (failover) are examples of resource attributes. The registration information also includes a path to an action script or application-specific action program that Oracle Clusterware calls to start, stop, check, and clean up the application.

An action script is a shell script that a generic script agent provided by Oracle Clusterware calls. An application-specific agent is usually a C or C++ program that calls Oracle Clusterware–provided APIs directly. Clusterware includes two script agents that make it possible to use scripts to protect an application. The two agents are:

- Script agent (`scriptagent` in Linux; `scriptagent.exe` in Windows): Use this agent to use shell or batch scripts to protect an application. Both the `cluster_resource` and `local_resource` resource types are configured to use this agent, and any resources of these types automatically take advantage of this agent.
- Application agent (`appagent` in Linux; `appagent.exe` in Windows): This agent automatically protects any resources of the application resource type used in previous versions of Oracle Clusterware. You are not required to configure anything to take advantage of the agent. It invokes action scripts in the manner done with previous versions of Clusterware and should be used only for the application resource type.

Agents

- An agent is a process that contains the agent framework and user code to manage resources.
- The agent framework is a library enabling you to plug in application-specific code to manage your applications.
- You program all of the management functions: starting, stopping, and checking the health of an application.
- These functions are referred to as entry points.
- The agent framework is responsible for invoking these entry point functions on behalf of Oracle Clusterware.
- Developers can use the entry points to plug in the required functionality for a specific resource.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware has access to application-specific primitives that have the ability to start, stop, and monitor a specific resource. Oracle Clusterware runs all resource-specific commands through an entity called an agent.

An agent is a process that contains the agent framework and user code to manage resources. The agent framework is a library that enables you to plug in your application-specific code to manage customized applications. You program all of the actual application management functions, such as starting, stopping, and checking the health of an application, into the agent. These functions are referred to as entry points.

The agent framework is responsible for invoking these entry point functions on behalf of Oracle Clusterware. Agent developers can use these entry points to plug in the required functionality for a specific resource regarding how to start, stop, and monitor a resource. Agents are capable of managing multiple resources. Agent developers can set entry points as callbacks to their code. These entry points include: ABORT, ACTION, CHECK, CLEAN, DELETE, MODIFY, START, and STOP. For more agent detail, refer to the *Oracle Clusterware Administration and Deployment Guide 12c Release 1 (12.1)*.

Action Scripts

- An action script defines one or more actions to start, stop, check, or clean a resource.
- The agent framework invokes these actions in the absence of the C/C++ actions.
- If all of the actions are defined in the script, then the script agent can invoke the actions defined in any action scripts.
- Before invoking the action script, the agent framework exports the necessary attributes from the resource profile.
- Resource attributes can be accessed from within an action script as environment variables prefixed with `_CRS_`.
 - For example, the `START_TIMEOUT` attribute becomes an environment variable named `_CRS_START_TIMEOUT`.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

An action script defines one or more actions to start, stop, check, or clean a resource. The agent framework invokes these actions in the absence of the C/C++ actions. Using action scripts, you can build an agent that contains the C/C++ entry points, as well as the script entry points. If all of the actions are defined in the action script, then you can use the script agent to invoke the actions defined in any action scripts.

Before invoking the action defined in the action script, the agent framework exports all the necessary attributes from the resource profile to the environment. Action scripts can log messages to the `stdout/stderr`, and the agent framework prints those messages in the agent logs. However, action scripts can use special tags to send the progress, warning, or error messages to the `crs*` client tools by prefixing one of the following tags to the messages printed to `stdout/stderr`:

`CRS_WARNING`
`CRS_ERROR`
`CRS_PROGRESS`

The agent framework strips out the prefixed tag when it sends the final message to the `crs*` clients. Resource attributes can be accessed from within an action script as environment variables prefixed with `_CRS_`. For example, the `START_TIMEOUT` attribute becomes an environment variable named `_CRS_START_TIMEOUT`.

Resource Types

- Oracle Clusterware uses resource types to organize resources employing similar attributes.
- Benefits from the use of resource types include:
 - You need to manage only necessary resource attributes.
 - You can manage all resources based on the resource type.
- All resources registered with Oracle Clusterware must be associated with a resource type.
- There are three resource types predefined in Oracle Clusterware:
 - Local resource
 - Cluster resource
 - Generic application



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Generally, resources are unique but some may have common attributes. Oracle Clusterware employs resource types to organize these similar resources. Benefits that resource types provide include:

- Manage only necessary attributes required by the resources.
- Easily manage many resources by managing the common resource type.

Every resource that you register in Oracle Clusterware must have a certain resource type. In addition to two resource types included in Oracle Clusterware, you can define custom resource types to suit your needs. The included resource types are:

- **Local resource:** These are server-centric resources; the type name is `local_resource`. These run locally on individual servers of the cluster and are not relevant outside of the scope of the server.
- **Cluster resource:** Cluster-aware resource types (type name is `cluster_resource`) are aware of the cluster environment and are subject to cardinality and cross-server switchover and failover.
- **Generic application:** You can use this resource type (type name is `generic_application`) to protect any generic applications without requiring additional scripts. High availability for an application is achieved by defining a resource with the `generic_application` resource type and providing the values for key attributes of the resource.

The `generic_application` resource type is derived from the `cluster_resource` resource type and, therefore, all resources of the `generic_application` resource type are cluster-aware resources. Attributes include:

- **START_PROGRAM:** A complete path to the executable that starts the application, with all appropriate arguments. The executable must exist on every server where Oracle Grid Infrastructure is configured to run the application.
- **STOP_PROGRAM:** A complete path to the executable that stops the application, with all appropriate arguments. The executable must exist on every server where Oracle Grid Infrastructure is configured to run the application. If you do not specify this attribute value, then Oracle Clusterware uses an operating system-equivalent of the `kill` command.
- **CLEAN_PROGRAM:** A complete path to the executable that cleans the program, with all appropriate arguments. The executable must exist on every server where Oracle Grid Infrastructure is configured to run the application. If you do not specify a value for this attribute, then Oracle Clusterware uses an operating system-equivalent of the `kill -9` command.
- **PID_FILES:** A comma-delimited list of complete paths to files that will be written by the application and contain a process ID (PID) to monitor. Failure of a single process is treated as a complete resource failure.
- **EXECUTABLE_NAMES:** A comma-delimited list of names of executables that is created when the application starts and the state of these executables is subsequently monitored. Failure of a single executable is treated as a complete resource failure.
- **ENVIRONMENT_FILE:** A complete path to the file containing environment variables to source when starting the application. The file must be a text file containing `name=value pairs`, one per line.
- **ENVIRONMENT_VARS:** A comma-delimited list of `name=value pairs` to be included into the environment when starting an application.
- **SEND_OUTPUT_ALWAYS:** This attribute is responsible for sending the application output that is sent to `STDOUT`, which is then displayed. A value of 0 does not display any application output except when an action fails. When an action fails, whatever application output that has been saved by the agent is displayed. Any value greater than 0 displays every application output. The default value is 0.

Adding Resource Types

- New resource types can be added using:
 - The `crsctl add type` command
 - Enterprise Manager
- Adding a resource type using `crsctl`:

```
crsctl add type my_resource_type -basetype cluster_resource      -attr
"ATTRIBUTE=PATH_NAME,TYPE=string,DEFAULT_VALUE=default.txt,
ATTRIBUTE=AGENT_FILENAME,TYPE=string,DEFAULT_VALUE=/path/to/agent"
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Custom resource types can be added using the `crsctl add type` command or Enterprise Manager Cloud Control. To add a resource type using `crsctl`, use the following syntax:

```
crsctl add type type_name -basetype base_type_name {-attr
"ATTRIBUTE=attribute_name | -file file_path,TYPE={string | int}
[,DEFAULT_VALUE=default_value] [,FLAGS=[READONLY] [|REQUIRED]]"}
```

Where:

type_name is a name for the resource type in the form of `xxx.yyy.type`.

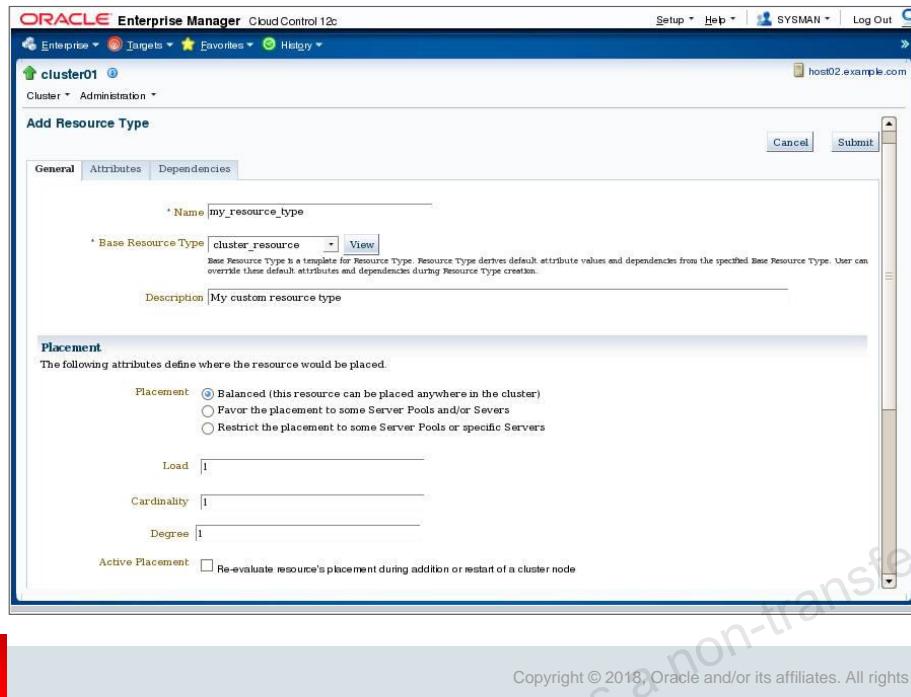
-basetype is the name of an existing base type. Any resource type that you create must either have `local_resource` or `cluster_resource` as its base resource type.

-attr is an attribute string. Each type attribute definition can contain up to four type attribute keywords that must be displayed in the order shown.

-file is a path name (either absolute or relative) for a text file containing line-delimited resource type keyword-value pairs that define the resource type.

DEFAULT_VALUE is the default value for the specified attribute.

Adding a Resource Type with EM



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

To add a new resource type to Oracle Clusterware by using Enterprise Manager, perform the following steps:

1. Log in to Oracle Enterprise Manager and click the Cluster link under Targets.
2. Click the Administration link to access the drop-down menu and select Resource Types, then click Add.
3. Enter a name for the resource type in the Name field.
4. Select either `cluster_resource`, `generic_application`, or `local_resource` as the base resource type from the Base Resource Type drop-down list. Generally, select the `cluster_resource` type for resources that can reside on any server in a cluster. Select the `local_resource` type for resources that must be present on each server of a cluster, by definition, such as VIPs, ASM instances, and network resources. Select `generic_application` for any generic applications without requiring additional scripts.
5. Define attributes controlling where the resource would be placed in the Placement section.
6. In the Action Program section of the page, you can define the way the resource type is started, stopped, and checked by Oracle Clusterware. Use the Action Program drop-down list to choose whether Oracle Clusterware calls an action script, an agent file, or both to manage the resource.

To further configure the resource type, click the Attributes folder tab. You can configure start, stop, and status attributes. You can also configure user-defined attributes. Click the Dependencies folder to configure start and stop dependencies between resources for the new resource type.

Using Clusterware to Enable High Availability

To manage your applications with Oracle Clusterware, perform the following steps:

1. Use the `generic_application` resource type, write a custom script for the script agent, or develop a new agent.
2. Register your applications as resources with Oracle Clusterware. Define resource attributes such as:
 - Path to the action script or application-specific agent
 - Privileges
 - Resource Dependencies
3. Assign the appropriate privileges to the resource.
4. Start or stop your resources.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Clusterware manages resources based on how you configure them to increase their availability. You can configure your resources so that Clusterware:

- Starts resources during cluster or server start
- Restarts resources when failures occur
- Relocates resources to other servers, if the servers are available

To manage your applications with Oracle Clusterware, perform the following steps:

1. Use the `generic_application` resource type, write a custom script for the script agent, or develop a new agent.
2. Register your applications as resources with Oracle Clusterware.
If a single application requires that you register multiple resources, you may be required to define relevant dependencies between the resources.
3. Assign the appropriate privileges to the resource.
4. Start or stop your resources.

When you register a resource in Oracle Clusterware, the relevant information about the application and the resource-relevant information is stored in the OCR.

This information includes:

- **Path to the action script or application-specific agent:** This is the absolute path to the script or application-specific agent that defines the start, stop, check, and clean actions that Oracle Clusterware performs on the application.
- **Privileges:** Oracle Clusterware has the necessary privileges to control all of the components of your application for high availability operations, including the right to start processes that are owned by other user identities. Oracle Clusterware must run as a privileged user to control applications with the correct start and stop processes.
- **Resource Dependencies:** You can create relationships among resources that imply an operational ordering or that affect the placement of resources on servers in the cluster. For example, Oracle Clusterware can only start a resource that has a hard start dependency on another resource if the other resource is running. Oracle Clusterware prevents stopping a resource if other resources that depend on it are running. However, you can force a resource to stop by using the `crsctl stop resource -f` command, which first stops all resources that depend on the resource being stopped.

Resource Attributes

- Read-only attributes:

```
ACTION_FAILURE_EVENT_TEMPLATE
INSTANCE_COUNT
INTERNAL_STATE
LAST_SERVER
LAST_STATE_CHANGE
PROFILE_CHANGE_EVENT_TEMPLATE
RESTART_COUNT
STATE
STATE_CHANGE_EVENT_TEMPLATE
STATE_DETAILS
TARGET
TARGET_SERVER
TYPE
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The resource attributes listed in the following are set by Clusterware when the resource is registered and are internally managed. You can view these attributes when you run `crsctl status resource` on a particular resource. These attributes include:

- **ACTION_FAILURE_EVENT_TEMPLATE**: An internal attribute for an `ora.*` resource
- **INSTANCE_COUNT**: Contains the number of instances that the resource currently has
- **INTERNAL_STATE**: Describes what, if any, action the policy engine is currently executing on the resource. Values include STARTING, STOPPING, CLEANING, and STABLE.
- **LAST_SERVER**: For `cluster_resource`-type resources, this attribute contains the name of the server on which the last start action for the resource succeeded. For `local_resource`-type resources, this is the name of the server to which the resource instance is pinned.
- **LAST_STATE_CHANGE**: Describes when the policy engine registers the current state of the resource
- **PROFILE_CHANGE_EVENT_TEMPLATE**: An internally managed attribute for an `ora.*` resource
- **RESTART_COUNT**: Used by the Clusterware daemon to count the number of attempts to restart a resource, starting from zero up to the value specified by `RESTART_ATTEMPTS`
- **STATE**: Reflects the current state of the resource as reported by Oracle Clusterware. Values include:
 - **ONLINE**: The resource is online and resource monitoring is enabled.
 - **OFFLINE**: The resource is offline and only offline resource monitoring is enabled.
 - **INTERMEDIATE**: The resource is either partially online or was known to be online before and subsequent attempts to determine its state have failed; resource monitoring is enabled.
 - **UNKNOWN**: The resource is unmanageable and its current state is unknown; manual intervention is required to resume its operation. A resource in this state is not monitored.

- **STATE_CHANGE_EVENT_TEMPLATE**: An internally managed attribute for an ora.* resource
- **STATE_DETAILS**: Details about the state of a resource. The four resource states: ONLINE, OFFLINE, UNKNOWN, and INTERMEDIATE may map to different resource-specific values, such as mounted, unmounted, and open. Agent developers can use the STATE_DETAILS attribute to provide a more detailed description of this mapping, resource to resource state.
- **TARGET**: Describes the desired state of a resource
- **TARGET_SERVER**: Contains the name of the server where the resource is starting
- **TYPE**: The type of resource indicated when you create a resource. This attribute is required when creating a resource and cannot be changed after the resource is created.

Resource Attributes

- Configurable attributes:

| ACL | DESCRIPTION | |
|------------------|----------------------------------|--------------------|
| ACTION_SCRIPT | ENABLED | SCRIPT_TIMEOUT |
| ACTION_TIMEOUT | FAILURE_INTERVAL | SERVER_POOLS |
| ACTIONS | FAILURE_THRESHOLD | SERVER_CATEGORY |
| ACTIVE_PLACEMENT | HOSTING_MEMBERS | START_CONCURRENCY |
| AGENT_FILENAME | INSTANCE_FAILOVER | START_DEPENDENCIES |
| ALERT_TEMPLATE | INTERMEDIATE_TIMEOUT | START_TIMEOUT |
| AUTO_START | LOAD | STOP_CONCURRENCY |
| CARDINALITY | MODIFY_TIMEOUT | STOP_DEPENDENCIES |
| CHECK_INTERVAL | NAME | STOP_TIMEOUT |
| CHECK_TIMEOUT | OFFLINE_CHECK_INTERVAL | UPTIME_THRESHOLD |
| CLEAN_TIMEOUT | RESTART_ATTEMPTS | USER_WORKLOAD |
| DELETE_TIMEOUT | PLACEMENT_RELOCATE_BY_DEPENDENCY | USE_STICKINESS |



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The resource attributes listed in the following are configurable with the `crsctl create|modify resource` command:

- ACL**: Defines the owner of a resource and the access privileges granted to various operating system users and groups
- ACTION_SCRIPT**: The path and file name of an action script
- ACTION_TIMEOUT**: Timeout value, in seconds, for all actions that Clusterware can perform on a resource
- ACTIONS**: Contains a space-delimited list of action specifications
- ACTIVE_PLACEMENT**: When set to 1, Clusterware uses this attribute to reevaluate the placement of a resource during addition or restart of a cluster server.
- AGENT_FILENAME**: Fully qualified file name of an agent program that a resource type uses to manage its resources
- ALERT_TEMPLATE**: Specifies additional resource attributes that are to be included in resource state alert messages
- AUTO_START**: Indicates whether Oracle Clusterware automatically starts a resource after a cluster server restart
- CARDINALITY**: Number of servers on which a resource can run, simultaneously
- CHECK_INTERVAL**: Time interval between repeated executions of the check action
- CHECK_TIMEOUT**: Maximum time, in seconds, in which a check action can run
- CLEAN_TIMEOUT**: Maximum time, in seconds, in which a clean action can run
- DELETE_TIMEOUT**: Maximum time, in seconds, in which a delete action can run
- DESCRIPTION**: A description of the resource you are adding
- ENABLED**: Clusterware uses this attribute to manage the state of the resource. Clusterware does not attempt to manage a disabled resource (`ENABLED=0`)

- **FAILURE_INTERVAL**: Interval, in seconds, before which Clusterware stops a resource
- **FAILURE_THRESHOLD**: Number of failures of a resource detected within a specified FAILURE_INTERVAL for the resource before Clusterware marks the resource as unavailable and no longer monitors it
- **HOSTING_MEMBERS**: Space-delimited, ordered list of server names that can host a resource
- **INSTANCE_FAILOVER**: Set to 0 to disallow the failover of resource instances from the servers on which they fail, binding the resource to a particular server
- **INTERMEDIATE_TIMEOUT**: Maximum amount of time in seconds that a resource can remain in the INTERMEDIATE state before the resource is declared as failed
- **LOAD**: Clusterware interprets the value of this attribute along with that of the PLACEMENT attribute. When the value of PLACEMENT is balanced, the value of LOAD determines where best to place a resource.
- **MODIFY_TIMEOUT**: Maximum time, in seconds, in which a modify action can run
- **NAME**: Case-sensitive alphanumeric string that names the resource
- **OFFLINE_CHECK_INTERVAL**: Controls offline monitoring of a resource
- **PLACEMENT**: Specifies how Clusterware selects a cluster server on which to start a resource
- **RESTART_ATTEMPTS**: Number of times that Clusterware attempts to restart a resource on the resource's current server before attempting to relocate it
- **RELOCATE_BY_DEPENDENCY**: Use to declare whether a resource will be enabled for relocation if requested to do so because of a dependency on the resource for which the relocation was requested
- **SCRIPT_TIMEOUT**: Maximum time (in seconds) for an action to run
- **SERVER_POOLS**: Space-delimited list of server pools to which a particular resource can belong
- **SERVER_CATEGORY**: For local resources, the definition of a local_resource type is extended to be category-aware
- **START_CONCURRENCY**: Maximum number of start actions that can be concurrent at a time
- **START_DEPENDENCIES**: A set of relationships that is considered when starting a resource
- **START_TIMEOUT**: Maximum time (in seconds) in which a start action can run
- **STOP_CONCURRENCY**: Maximum number of stop actions that can be concurrent at a time
- **STOP_DEPENDENCIES**: A set of relationships that is considered when stopping a resource
- **STOP_TIMEOUT**: Maximum time (in seconds) in which a stop or clean action can run
- **UPTIME_THRESHOLD**: Length of time that a resource must be up before Clusterware considers the resource to be stable
- **USER_WORKLOAD**: Indicates if a resource is a workload-generating resource for what-if analysis
- **USE_STICKINESS**: Use to indicate that a resource should run where it last ran, if possible

Resource States

| State | Description |
|--------------|---|
| ONLINE | The resource is running. |
| OFFLINE | The resource is not running. |
| UNKNOWN | An attempt to stop the resource has failed. Clusterware does not actively monitor resources that are in this state. |
| INTERMEDIATE | A resource can be in the INTERMEDIATE state because of one of two events: 1. Clusterware cannot determine the state of the resource but the resource was either attempting to go online or was online the last time its state was precisely known. 2. A resource is partially online. Clusterware actively monitors resources that are in the INTERMEDIATE state. Clusterware transitions the resource out of the INTERMEDIATE state automatically as soon as it is appropriate. |



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Every resource in a cluster is in a particular state at any time. Certain actions or events can cause that state to change. The table above lists and describes the possible resource states.

Resource Dependencies

- You can configure resources to be dependent on other resources.
 - Dependent resources will only start or stop when certain conditions of the resources on which they depend are met.
- You can configure resources so that they depend on Oracle resources.
- Resource dependencies are separated into start and stop categories.
- This separation improves and expands the start and stop dependencies between resources and resource types.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

You can configure resources to be dependent on other resources, so that the dependent resources can only start or stop when certain conditions of the resources on which they depend are met. For example, when Clusterware attempts to start a resource, it is necessary for any resources on which the initial resource depends to be running and in the same location. If Oracle Clusterware cannot bring the resources online, then the initial (dependent) resource cannot be brought online, either. If Clusterware stops a resource or a resource fails, then any dependent resource is also stopped.

Some resources require more time to start than others. Some resources must start whenever a server starts, whereas other resources require a manual start action. These and many other examples of resource-specific behavior imply that each resource must be described in terms of how it is expected to behave and how it relates to other resources (resource dependencies). You can configure resources so that they depend on Oracle resources. When creating resources, however, do not use an `ora` prefix in the resource name. This prefix is reserved for Oracle use only.

Previous versions of Oracle Clusterware included only two dependency specifications: the `REQUIRED_RESOURCES` resource attribute and the `OPTIONAL_RESOURCES` resource attribute. The `REQUIRED_RESOURCES` resource attribute applied to both start and stop resource dependencies. Resource dependencies are separated into start and stop categories. This separation improves and expands the start and stop dependencies between resources and resource types.

Start Dependencies

You can configure the attraction start dependency with the following constraints:

Hard: Defines a hard start dependency for a resource if another resource must be `ONLINE` before the dependent resource can start

Weak: Indicates an attempt is made to start the resource on which the resource in question is dependent on if it is not `ONLINE`

Attraction: Indicates that Clusterware will attempt to start a resource on the same node on which the resource it is dependent on is running

Pullup: When the `pullup` dependency is set for a resource, the resource starts as a result of starting the named resources

Dispersion: Indicates that the resource will not be located on the same server as dependent resources, if possible



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Clusterware considers start dependencies contained in the profile of a resource when the start effort evaluation for that resource begins. You specify start dependencies for resources using the `START_DEPENDENCIES` resource attribute. You can use modifiers on each dependency to further configure the dependency.

Configure the `attraction` start dependency with the following constraints:

- `START_DEPENDENCIES=attraction(intermediate:resourceB)`
Use the `intermediate` modifier to specify whether the resource is attracted to resources that are in the `INTERMEDIATE` state.
- `START_DEPENDENCIES=attraction(type:resourceB.type)`
Use the `type` modifier to specify if the dependency acts on a particular resource type.

Note: Previous versions of Clusterware used the now deprecated `OPTIONAL_RESOURCES` attribute to express attraction dependency.

You can configure the `dispersion` start dependency with the following modifiers:

- `START_DEPENDENCIES=dispersion(intermediate:resourceB)`
Use the `intermediate` modifier to specify that Oracle Clusterware disperses resource A and whether resource B is either in the `ONLINE` or `INTERMEDIATE` state.
- `START_DEPENDENCIES=dispersion:active(resourceB)`
Typically, dispersion is only applied when starting resources.
- `START_DEPENDENCIES=dispersion(pool:resourceB)`
Use the `pool` modifier to specify that Oracle Clusterware disperses the resource to a different server pool rather than to a different server.

You can configure the **exclusion** start dependency with the following modifiers:

- START_DEPENDENCIES=exclusion([[preempt_pre: | preempt_post:]] target_resource_name | type:target_resource_type]*)
 - preempt_pre:** CRSD stops the specified target resource or resources defined by a specific resource type before starting the source resource.
 - preempt_post:** CRSD stops and relocates, if possible, the specified target resource or resources defined by a specific resource type.

You can configure the **hard** start dependency with the following constraints:

- START_DEPENDENCIES=hard(global:resourceB)
 - By default, resources A and B must be located on the same server (collocated). Use the global modifier to specify that resources need not be collocated.
- START_DEPENDENCIES=hard(intermediate:resourceB)
 - Use the intermediate modifier to specify that the dependent resource can start if a resource on which it depends is in either the ONLINE or INTERMEDIATE state.
- START_DEPENDENCIES=hard(type:resourceB.type)
 - Use the type modifier to specify whether the hard start dependency acts on a particular resource or a resource type.
- START_DEPENDENCIES=hard(uniform:resourceB)
 - Use the uniform modifier to attempt to start all instances of resource B, but only one instance, at least must start to satisfy the dependency.
- START_DEPENDENCIES=hard(resourceB, intermediate:resourceC, intermediate:global:type:resourceC.type)
 -

You can configure the **pullup** start dependency with the following constraints:

- START_DEPENDENCIES=pullup(intermediate:resourceB)
 - Use the intermediate modifier to specify whether resource B can be either in the ONLINE or INTERMEDIATE state to start resource A.
- START_DEPENDENCIES=pullup:always(resourceB)
 - Use the always modifier to specify whether Oracle Clusterware starts resource A despite the value of its TARGET attribute, whether it is ONLINE or OFFLINE.
- START_DEPENDENCIES=pullup(type:resourceB.type)
 - Use the type modifier to specify that the dependency acts on a particular resource type.

You can configure the **weak** start dependency with the following constraints:

- START_DEPENDENCIES=weak(global:resourceB)

By default, resources A and B must be collocated. Use the global modifier to specify that resources need not be collocated.

START_DEPENDENCIES=weak(concurrent:resourceB)

Use the concurrent modifier to specify that resources A and B can start concurrently, instead of waiting for resource B to start first.

START_DEPENDENCIES=weak(type:resourceB.type)

Use the type modifier to specify that the dependency acts on a resource of a particular resource type.

START_DEPENDENCIES=weak(uniform:resourceB)

Use the uniform modifier to attempt to start all instances of resource B.

Stop Dependencies

- Clusterware considers stop dependencies whenever a resource changes from `ONLINE` to any other state.
- The only value for `STOP_DEPENDENCIES` is `hard`.
- You can configure the `hard` stop dependency with the following modifiers:

```
STOP_DEPENDENCIES=hard(intermediate:resourceB)
STOP_DEPENDENCIES=hard(global:resourceB)
STOP_DEPENDENCIES=hard(shutdown:resourceB)
```

- Use `intermediate` to specify whether resource B must be either `ONLINE` or `INTERMEDIATE` for resource A to be online.
- Use `global` to specify if resource A requires B be present on the same server or on any cluster server to remain online.
- Use the `shutdown` modifier to stop the resource only when you shut down the Clusterware stack with `crsctl`.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware considers stop dependencies between resources whenever a resource is stopped (the resource state changes from `ONLINE` to any other state).

If resource A has a `hard` stop dependency on resource B, then resource A must be stopped when B stops running. The two resources may attempt to start or relocate to another server, depending upon how they are configured. Oracle recommends that resources with `hard` stop dependencies also have `hard` start dependencies.

You can configure the `hard` stop dependency with the following modifiers:

- `STOP_DEPENDENCIES=hard(intermediate:resourceB)`

Use the `intermediate` modifier to specify whether resource B must be in either the `ONLINE` or `INTERMEDIATE` state for resource A to stay online.

- `STOP_DEPENDENCIES=hard(global:resourceB)`

Use the `global` modifier to specify whether resource A requires that resource B be present on the same server or on any server in the cluster to remain online. If this constraint is not specified, then resources A and B must be running on the same server. Oracle Clusterware stops resource A when that condition is no longer met.

- `STOP_DEPENDENCIES=hard(shutdown:resourceB)`

Use the `shutdown` modifier to stop the resource only when you shut down the Clusterware stack by using either the `crsctl stop crs` or `crsctl stop cluster` command.

Creating a Clusterware Managed Application VIP

- If clients access the application through a network, you must register a VIP on which the application depends.
- An application VIP is a cluster resource that Oracle Clusterware manages.
- Oracle recommends using the `appvipcfg` utility to create or delete an application VIP on the default network (1).
- To create an application VIP on the default network:

```
# appvipcfg create -network=1 -ip=192.0.2.170  
-vipname=appsVIP -user=root
```

- To view the new VIP profile:

```
# crsctl status res appsVIP -p  
NAME=appsVIP  
TYPE=app.appvip_net1.type  
ACL=owner:root:rwx,pgrp:root:r-x,other::r--,user:root:r-x  
...
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

If clients of an application access the application through a network, and the placement policy for the application allows it to fail over to another node, then you must register a virtual internet protocol address (VIP) on which the application depends. An application VIP is a cluster resource that Oracle Clusterware manages (Oracle Clusterware provides a standard VIP agent for application VIPs). You should base any new application VIPs on this VIP type to ensure that your system experiences consistent behavior among all of the VIPs that you deploy in your cluster.

Although you can add a VIP in the same way that you can add any other resource that Oracle Clusterware manages, Oracle recommends using the `Grid_home/bin/appvipcfg` command-line utility to create or delete an application VIP on the default network for which the `ora.net1.network` resource is created by default.

To create an application VIP, use the following syntax:

```
appvipcfg create -network=network_number -ip=ip_address -vipname=vip_name -  
user=user_name [-group=group_name] [-failback=0|1]
```

The `appvipcfg` script requires that you specify the `-network` option, even if `-network=1`.

After you have created the application VIP using this configuration script, you can view the VIP profile using the following command:

```
crsctl status res appsVIP -p
```

As the Oracle Database installation owner, start the VIP resource:

```
$ crsctl start resource appsVIP
```

To delete an application VIP, use the `appvipcfg` script with the `delete` option. This option accepts the VIP name as a parameter. For example:

```
# appvipcfg delete -vipname=appsVIP
Production Copyright 2007, 2008, Oracle. All rights reserved
2013-07-19 20:11:50: Deleting the resource
2013-07-19 20:11:50: Executing /u01/app/12.1.0/grid/bin/crsctl delete res
appsVIP
2013-07-19 20:11:50: Executing cmd: /u01/app/12.1.0/grid/bin/crsctl delete
res appsVIP
2013-07-19 20:11:50: Removing the type
2013-07-19 20:11:50: Executing /u01/app/12.1.0/grid/bin/crsctl delete type
app.appvip_net1.type

2013-07-19 20:11:50: Executing cmd: /u01/app/12.1.0/grid/bin/crsctl delete
type app.appvip_net1.type
```

Creating an Application VIP Using EM

The first screenshot shows the 'Add Application VIP Resource' dialog. It has fields for Name (MyAppVip), Network Number (1), Internet Protocol Address (192.0.2.170), and Primary User (root). A checkbox 'Start the resource after creation' is checked. Buttons 'Cancel' and 'Continue' are at the bottom.

The second screenshot shows the 'Confirmation: Add VIP Resource' dialog. It displays the command being run: 'appvipcfg create -network=1 ip=192.0.2.170 -vipname=MyAppVip -user=root'. It asks if the user wants to proceed with superuser privileges. Below it is a 'Specify Cluster Credentials' dialog with 'Credential' set to 'New', 'UserName' as 'root', and 'Password' as 'root'. A 'Save As' option is available. Buttons 'Cancel', 'Continue', and 'Finish' are at the bottom.

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

To create an application VIP with Oracle Enterprise Manager, perform the following steps:

1. Log in to Oracle Enterprise Manager Cloud Control.
2. Select the cluster target that you want to modify.
3. From the cluster target menu, select Administration > Resources > Manage.
4. Enter a cluster administrator username and password to display the Manage Resources page.
5. Click Add Application VIP.
6. Enter a name for the VIP in the Name field.
7. Enter a network number in the Network Number field.
8. Enter an IP address for the VIP in the Internet Protocol Address field.
9. Enter `root` in the Primary User field. Oracle Enterprise Manager defaults to whatever username you are logged in as.
10. Select “Start the resource after creation” if you want the VIP to start immediately.
11. Click Continue to display the Confirmation: Add VIP Resource page.
12. Enter `root` and the `root` password as the cluster credentials.
13. Click Continue to create the application VIP.

Deciding on a Deployment Scheme

- You must decide whether to use administrator or policy management for the application resource.
- Use administrator management for smaller configurations where your cluster configuration is not likely to change.
- Use policy management for more dynamic configurations when your cluster consists of more than two nodes.
- A cluster hosting applications deployed in both of schemes can be viewed as two logically separate groups of servers.
 - One server group can be used for server pools, enabling role separation and server capacity control.
 - Another server group assumes a fixed assignment based on named servers in the cluster.
- The Generic pool always owns the servers used by applications of administrator-based management.

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Registering a Resource

- Register resources in Oracle Clusterware 12c using the `crsctl add resource` command.

```
$ crsctl add resource resource_name -type res_type [-file file_path] | [-attr "attribute_name='value', attribute_name='value', ..."]
```

- Structure of a sample attribute file:

```
PLACEMENT=favored
HOSTING_MEMBERS=host01 host02 host03
RESTART_ATTEMPTS@CARDINALITYID(1)=0
RESTART_ATTEMPTS@CARDINALITYID(2)=0
FAILURE_THRESHOLD@CARDINALITYID(1)=2
FAILURE_THRESHOLD@CARDINALITYID(2)=4
FAILURE_INTERVAL@CARDINALITYID(1)=300
FAILURE_INTERVAL@CARDINALITYID(2)=500
CHECK_INTERVAL=2
CARDINALITY=2
```

- Using the `-attr` option:

```
$ crsctl add resource MyResource -type cluster_resource [-attr "PLACEMENT='favored',
HOSTING_MEMBERS='node1 node2 node3', ..."]
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

You can register resources in Oracle Clusterware 12c by using the `crsctl add resource` command. To register an application as a resource, use the following syntax:

```
$ crsctl add resource resource_name -type resource_type [-file file_path] | [-attr "attribute_name='attribute_value', attribute_name='attribute_value', ..."]
```

Choose a name for the resource based on the application for which it is being created. For example, if you create a resource for an Apache web server, then you might name the resource `myApache`.

The name of the resource type follows the `-type` option. You can specify resource attributes in either a text file specified with the `-file` option or in a comma-delimited list of resource attribute-value pairs enclosed in double quotation marks ("") following the `-attr` option. You must enclose space or comma-delimited attribute names and values enclosed in parentheses in single quotation marks ("").

Registering a Resource: Example

- To register the Apache web server as:
 - An administrator-managed resource:

```
# crsctl add resource myApache -type cluster_resource -attr
"ACTION_SCRIPT='/u01/ogi/scripts/myapache.scr', PLACEMENT='restricted',
HOSTING_MEMBERS='host01 host02',
CHECK_INTERVAL='30',START_DEPENDENCIES='hard(appsvip)',
STOP_DEPENDENCIES='hard(appsvip)', RESTART_ATTEMPTS='2',"
```

- A policy-managed resource:

```
# crsctl add resource myApache -type cluster_resource -attr
"ACTION_SCRIPT='/u01/ogi/scripts/myapache.scr', PLACEMENT='restricted',
SERVER_POOLS='myServerPool', CHECK_INTERVAL='30', START_DEPENDENCIES='hard(appsvip)',
STOP_DEPENDENCIES='hard(appsvip)',RESTART_ATTEMPTS='2',"
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

To register an application using `crsctl`, determine whether it will be administrator or policy-managed. The example above shows how to add an application as either an administrator-managed resource or a policy-managed resource. You can specify an administrator-managed resource by defining the `HOSTING_MEMBERS` parameter or by defining a server pool as a subpool of the Generic pool. For example:

```
$ crsctl add serverpool myApache_sp -attr "PARENT_POOLS=Generic,
SERVER_NAMES=host36 host37"
```

After you create the subpool, add the Apache web server resource, as follows:

```
$ crsctl add resource myApache -type cluster_resource -attr
"ACTION_SCRIPT='/u01/ogi/scripts/myapache.scr,
PLACEMENT='restricted',
SERVER_POOLS=myApache_sp,
CHECK_INTERVAL='30',
RESTART_ATTEMPTS='2',
START_DEPENDENCIES='hard(appsvip)',
STOP_DEPENDENCIES='hard(appsvip)'"
```

To add the Apache web server to a specific server pool as a resource using the policy-based deployment scheme, run the following command as the user that is supposed to run the Apache Server (typically `root`):

```
$ crsctl add resource myApache -type cluster_resource -attr
"ACTION_SCRIPT=/opt/cluster/scripts/myapache.scr,
PLACEMENT=restricted,
SERVER_POOLS=server_pool_list,
CHECK_INTERVAL=30,
RESTART_ATTEMPTS=2,
START_DEPENDENCIES=hard(appsvip),
STOP_DEPENDENCIES=hard(appsvip)"
```

Adding Resources with EM

The screenshot shows the Oracle Enterprise Manager Cloud Control 12c interface. The top navigation bar includes 'Enterprise', 'Targets', 'Favorites', 'History', 'Setup', 'Help', 'SYSMAN', and 'Log Out'. The main title is 'cluster01'. The 'Administration' menu is open, and the 'Resources' option is selected. The 'Manage' sub-menu is open, and the 'Add' option is selected. The 'General' tab is active. The 'Name' field contains 'MyResource'. The 'Resource Type' dropdown is set to 'cluster_resource'. Below it, a note states: 'Resource Type is a template for the Resource. The Resource derives default attribute values and dependencies from the specified Resource Type. The user can override these default attributes and dependencies during Resource creation.' The 'Description' field contains 'My application resource'. A checkbox labeled 'Start the resource after creation' is checked. The 'Placement' section has the following configuration: 'Placement' is set to 'Balanced' (radio button selected), 'Load' is 1, 'Cardinality' is 1, and 'Degree' is 1. The bottom right corner of the interface has a watermark: 'ANGEL (angelbabu2000@gmail.com) has a non-transferable license to use this material.'

ORACLE®

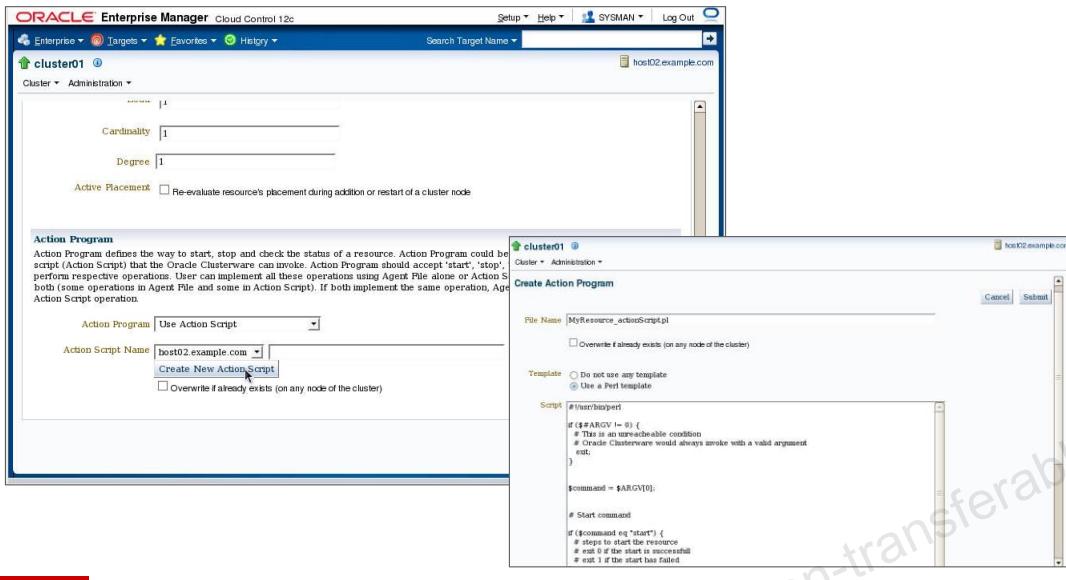
Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

To add resources to Oracle Clusterware by using Oracle Enterprise Manager, perform the following steps:

1. Log in to Oracle Enterprise Manager Cloud Control. Select the cluster target that you want to modify.
2. From the cluster target menu, select Administration > Resources > Manage.
3. Enter a cluster administrator username and password to display the Add Resource page.
4. Enter a name for the resource in the Name field.
5. Choose either `cluster_resource` or `local_resource` from the Resource Type drop-down list.
6. Oracle Clusterware uses resource types to organize these similar resources. Every resource that you register in Oracle Clusterware must have a certain resource type. In addition to two resource types included in Oracle Clusterware, you can define custom resource types. Generally, select the `cluster_resource` type for resources that can reside on any server in a cluster. Select the `local_resource` type for resources that must be present on each server of a cluster, by definition, such as VIPs, ASM instances, and network resources.
7. Select “Start the resource after creation” if you want the resource to start immediately.
8. The optional parameters in the Placement section define where in a cluster Oracle Clusterware places the resource.

Adding Resources with EM

Resource Parameters



ORACLE®

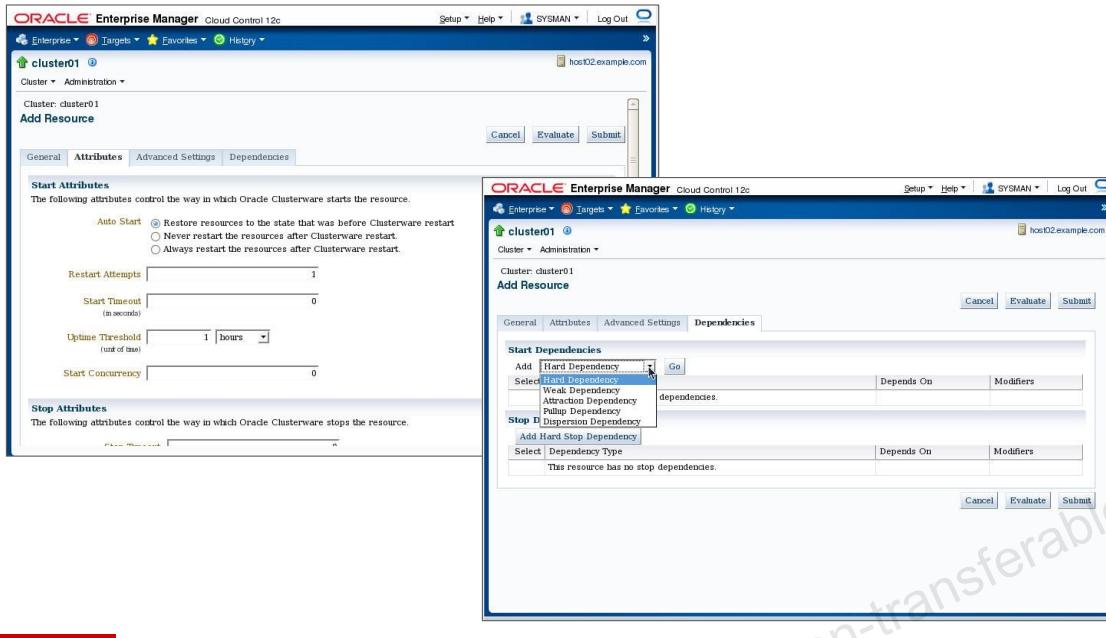
Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

10. In the Action Program section, choose from the Action Program drop-down list whether Oracle Clusterware calls an action script, an agent file, or both to manage the resource.

You must also specify a path to the script, file, or both, depending on what you select from the drop-down list.

If you choose Action Script, then you can click Create New Action Script to use the Oracle Enterprise Manager action script template to create an action script for your resource, if you have not yet done so.

Adding Resources with EM



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

11. To further configure the resource, click Attributes. On this page, you can configure start, stop, and status attributes. In addition, you can configure offline monitoring behavior and any user-defined attributes that you wish to implement.
12. Click Advanced Settings to enable more detailed resource attribute configurations.
13. Click Dependencies to configure start and stop dependencies between resources.
14. Click Submit when you finish configuring the resource.

Managing Resources with crsctl

- Start or stop resources with `crsctl start|stop resource`:

```
# crsctl stop resource myApache
CRS-2673: Attempting to stop 'myApache' on 'host01'
CRS-2677: Stop of 'myApache' on 'host01' succeeded
# crsctl start resource myApache
CRS-2672: Attempting to start 'myApache' on 'host01'
CRS-2676: Start of 'myApache' on 'host01' succeeded
```

- Use the `crsctl relocate resource` command to relocate applications and application resources:

```
# crsctl relocate resource myApache -n host02
CRS-2673: Attempting to stop 'myApache' on 'host01'
CRS-2677: Stop of 'myApache' on 'host01' succeeded
CRS-2672: Attempting to start 'myApache' on 'host02'
CRS-2676: Start of 'myApache' on 'host02' succeeded
```

- To unregister a resource:

```
# crsctl delete resource myApache
```

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Start and stop resources with the `crsctl start|stop resource` commands. Manual starts or stops outside of Oracle Clusterware can invalidate the resource status. In addition, Oracle Clusterware may attempt to restart a resource on which you perform a manual stop operation. Running `crsctl start resource` on a resource sets the resource target value to `ONLINE`. Clusterware attempts to change the state to match the target by running the action program with the `start` parameter. When a resource is running, both the target state and the current state are `ONLINE`.

To start an application resource that is registered with Oracle Clusterware, use the `crsctl start resource` command. For example:

```
# crsctl start resource myApache
```

Use the `crsctl relocate resource` command to relocate applications and application resources. To relocate the Apache web server application to a server named `host02`, run the following command:

```
# crsctl relocate resource myApache -n host02
```

To relocate an application and its required resources, use the `-f` option with the `crsctl relocate resource` command.

Run `crsctl delete resource` as a clean-up step when a resource is no longer managed by Oracle Clusterware. It is recommended that you unregister any unnecessary resources.

Managing Resources with crsctl

Enter the following command to view information about all applications and resources.

```
# crsctl status resource
NAME=ora.DATA.dg
TYPE=ora.diskgroup.type
TARGET=ONLINE          , ONLINE
STATE=ONLINE on host01, ONLINE on host02

NAME=ora.LISTENER.lsnr
TYPE=ora.listener.type
TARGET=ONLINE          , ONLINE
STATE=ONLINE on host01, ONLINE on host02
...
NAME=ora.rdba.db
TYPE=ora.database.type
TARGET=ONLINE          , ONLINE
STATE=ONLINE on host01, ONLINE on host02

NAME=ora.scan3.vip
TYPE=ora.scan_vip.type
TARGET=ONLINE
STATE=ONLINE on host02
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

To display status information about applications and resources that are on cluster servers, use the `crsctl status resource` command. The following example displays the status information for the Apache web server application:

```
# crsctl status resource myApache
NAME=myApache
TYPE=cluster_resource
TARGET=ONLINE
STATE=ONLINE on host01
```

Enter the following command to view information about all applications and resources in tabular format:

```
# crsctl status resource
```

Append a resource name to the preceding command to determine:

- How many times the resource has been restarted
- How many times the resource has failed within the failure interval
- The maximum number of times that a resource can restart or fail
- The target state of the resource and the normal status information

Use the `-f` option with the `crsctl status resource resource_name` command to view full information about a specific resource.

Managing Clusterware Resources with EM

Management actions

The screenshot shows the Oracle Enterprise Manager interface for managing Clusterware resources. The top navigation bar includes links for Enterprise, Targets, Favorites, History, Setup, Help, SYSMAN, and Log Out. The main content area is titled 'Resources' and displays a list of 23 resources, including Oracle resources. The toolbar below the list contains buttons for View, Edit, Remove, Start, Stop, Relocate, Evaluate, and Add. A red box highlights this toolbar, and a red arrow points from the text 'Management actions' to it. The resource list table includes columns for Select, Details, Name, Cardinality, Current State, Target State, Running Hosts, Resource Type, and Owner. The first five resources listed are: ora.MGMTLSNR, ora.mgmtdb, ora.DATA.dg, ora.PRA.dg, and ora.LISTENER.lsnr. All resources show a current state of 'Down' and a target state of 'Down'. The running hosts for ora.DATA.dg, ora.PRA.dg, and ora.LISTENER.lsnr are host01, host02, and host03 respectively. The resource types are ora.mgmtlsnr.type, ora.mgmtdb.type, ora.diskgroup.type, ora.diskgroup.type, and ora.listener.type. The owner is 'grid' for all resources.

| Select | Details | Name | Cardinality | Current State | Target State | Running Hosts | Resource Type | Owner |
|--------------------------|---------|-------------------|---------------------|---------------|--------------|----------------------|--------------------|-------|
| <input type="checkbox"/> | > Show | ora.MGMTLSNR | 1 | Down | Down | n/a | ora.mgmtlsnr.type | grid |
| <input type="checkbox"/> | > Show | ora.mgmtdb | 1 | Down | Down | n/a | ora.mgmtdb.type | grid |
| <input type="checkbox"/> | > Show | ora.DATA.dg | Runs on all servers | Up | Up | host01,host02,host03 | ora.diskgroup.type | grid |
| <input type="checkbox"/> | > Show | ora.PRA.dg | Runs on all servers | Up | Up | host01,host02,host03 | ora.diskgroup.type | grid |
| <input type="checkbox"/> | > Show | ora.LISTENER.lsnr | Runs on all servers | Up | Up | host01,host02,host03 | ora.listener.type | grid |

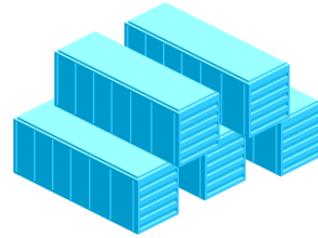
You can use Oracle Enterprise Manager to manage Oracle Clusterware resources. You can create and configure resources in Oracle Clusterware and also monitor and manage resources after they are deployed in the cluster. Resource Management tasks that can be performed from EM include:

- Editing resource attributes
- Removing resources
- Starting and stopping resources
- Relocating a resource

Using Oracle Enterprise Manager to monitor and manage various Oracle Clusterware resources eases the daily management in high availability environments.

Clusterware Resource Groups

- A resource group is a container for a logically related group of resources.
- An application is modeled as a resource group that contains:
 - The application resource
 - Related application resources such as WebServer
 - Infrastructure resources such as disk groups and VIPs
- A resource group provides a logical and intuitive entity for high availability modeling of all classes of applications.
- Resource groups are created and resources are added to the resource groups by using `crsctl`.
- Attributes define naming, description, and common placement and failover parameter values for the resource.



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

A resource group is a container for a logically related group of resources. An application is modeled as a resource group that contains the application resource, related application resources (such as WebServer), and infrastructure resources (such as disk groups and VIPs). A resource group provides a logical and intuitive entity for high availability modeling of all classes of applications.

You create resource groups by using `crsctl`, and then add resources to the resource group. A resource group provides a set of attributes that define naming, description, and common placement and failover parameter values for the resources that are members of the resource group.

Resource Group: Overview

- You create a resource group based on a resource group type.
- A resource can be a member of only one resource group.
- If a resource group is not given when a resource is created, it becomes a member of an automatic resource group that is created for that resource.
- Resource groups are aware of critical resources and the state of the resource group is determined by the state of its critical resources.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

You create a resource group based on a resource group type. A resource can be a member of only one resource group. You can specify a resource group for a resource when you create the resource. If you do not specify a resource group when you create a resource, the resource becomes a member of an automatic resource group that is created for that resource. You can later add the resource to a different resource group.

Resource groups are aware of critical resources and the state of the resource group is solely determined by the state of its critical resources. You can remove a non-critical resource from a resource group (subject to dependency checks) and, at a later time, add it to another resource group.

Resource groups have cardinality to specify the number of instances of the resource group that can simultaneously run in the cluster.

All member resources of a running resource group instance are located on the same server. Oracle Clusterware restarts a resource group in the event of failure, and then relocates the resource group to another server in the event of local restart failures.

Automatic Resource Groups

- An automatic resource group is created for each resource that is not explicitly added to a resource group.

```
$ crsctl add resource my_apache -type generic_application ...
$ crsctl stat resourcegroup my_apache -f | grep CRITICAL
CRITICAL_RESOURCES=my_apache
```

- Resources that you create without specifying a resource group can be added to a resource group at a later time.

```
$ crsctl modify resource my_apache -group rg1
$ crsctl stat resourcegroup rg1 -f | grep CRITICAL
CRITICAL_RESOURCES=my_apache
```

- Clusterware deletes the automatic resource group when the resource is explicitly added to a resource group.

```
$ crsctl stat resourcegroup my_apache -f
CRS-33613: could not find resource group 'my_apache'
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

If you create a resource without specifying a resource group, Oracle Clusterware implicitly and automatically adds the resource to a resource group with the same name as the resource.

An automatic resource group is created for each resource that is not explicitly added to a resource group. You can create resources without using resource groups and work with Oracle Clusterware without disruption. Using resource groups, however, enables you to define relationships to the infrastructure and application resources (through automatic resource groups) that are created by SRVM or other existing utilities.

An automatic resource group is solely described by the resource that it has been created for, and cannot be modified by an administrator. Resources that you create without specifying a resource group can be added to a resource group at a later time. Oracle Clusterware deletes the automatic resource group to which the resource belongs when the resource is explicitly added to a resource group.

Resource Group Privileges

- You can set privileges for modifying and executing operations on a resource group by setting the ACL attribute.
- The resource group owner can assign privileges to other OS users and groups by appropriately setting the ACL attribute.
- A resource within a resource group can maintain its own privilege specification within its ACL attribute:
 - A user with write privilege on a resource group and write privilege on a resource can add the resource to the group.
 - The owner of the resource group must, at all times, have execute privileges on all resources in the group.
 - Any user or group granted execute privileges on the group must have execute privileges on all resources in the group.
 - The local administrative user can modify, delete, start, and stop any resource group.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

You can create resource groups and resource group types, and then create and add resources to these groups. You can also define privileges for modifying and executing operations on a resource group by using the ACL attribute of the resource group. The resource group owner can assign privileges to other operating system users and groups by appropriately setting the ACL attribute of the resource group. A resource within a resource group can maintain its own privilege specification within its ACL attribute. Specifically:

- A user with write privilege on a resource group and write privilege on a resource can add the resource to the group
- The owner of the resource group must, at all times, have execute privileges on all the resources in the group. Any user or group granted execute privileges on a group must have execute privileges on all the resources in the group. For example, in cases where certain infrastructure resources in a resource group must be managed by `root`, the owner of the resource must be specified as `root` and have execute permissions on the resource granted to the group owner. This must be done explicitly by the root user.
- The local administrative user (root on UNIX or Administrators group user on Windows) can modify, delete, start, and stop any resource group.

Resource Group Dependencies

- Dependencies can be set among resource groups, allowing a way to define relationships between applications and components.
- Clusterware provides modifiers to specify different ordering, location, and enforcement level of dependencies among resource groups.
- All available Oracle Clusterware resource dependencies are also available to use with resource groups.
- The `START_DEPENDENCIES` and `STOP_DEPENDENCIES` attributes of a resource group can be defined to specify dependencies for resource groups.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

You can set dependencies among resource groups, thus providing a means to express relationships between applications and components. Clusterware provides modifiers to specify different ordering, location, and enforcement level of dependencies among resource groups. Some things to consider about resource group dependencies:

- A resource group can have a dependency relationship with another resource group and not to individual resources.
- An explicitly created resource group can have a dependency relationship with an automatic resource group.
- A resource in a group can have a dependency relationship with another resource in the same group.
- Resources that are created without specifying a resource group (thus belonging to an automatic resource group) can have a dependency relationship with another resource group.
- A resource cannot have a dependency relationship with a resource group or a resource in a different resource group.

All available Oracle Clusterware resource dependencies are also available for use with resource groups. You configure the `START_DEPENDENCIES` and `STOP_DEPENDENCIES` attributes of a resource group to specify dependencies for resource groups.

Resource Group Dependency Types and Modifiers

| Dependency Type | Description |
|-----------------|--|
| hard start | Specifies specific other resource groups that must be online anywhere in the cluster before this resource group can be started |
| weak start | Specifies the requirement that an attempt must be made to start specific other resource groups before starting this resource group |
| pullup | This dependency is used when the resource group must be automatically started when a dependent resource group starts. |
| hard stop | Specifies the mandatory requirement of stopping this resource group when another specific resource group stops running |
| attraction | Specifies a co-location preference with other resource groups |
| dispersion | Specifies a preference to not be co-located with specific other resource groups |
| exclusion | Specifies a mandatory requirement that this resource group should not run on the same server as specific other resource groups |



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Failure and Recovery of Critical Resources

1. When a critical resource of a resource group fails, the resource group immediately transitions to OFFLINE.
2. Clusterware attempts a local restart of the failed resource according to the RESTART_ATTEMPTS and UPTIME_THRESHOLD attributes.
3. Clusterware initiates immediate check actions on other resources in the same group that have a stop dependency on the failed resource.
4. Clusterware initiates immediate check actions on other resource groups that are dependent on this resource group.
5. If the resource restarts successfully, the resource group transitions to the ONLINE state.
 - Pullup dependency is evaluated within and across resource groups.
6. If Clusterware exhausts all local restart attempts of the resource, it stops the entire resource group.
 - On exhausting all restart attempts, the resource group will fail over to another server.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Critical resources determine resource group state and failover. When a critical resource of a resource group fails, the resource group immediately transitions to the OFFLINE state. Oracle Clusterware attempts a local restart of the failed critical resource according to the RESTART_ATTEMPTS and UPTIME_THRESHOLD resource attributes.

Oracle Clusterware initiates immediate check actions on other resources in the same group that have a stop dependency on the failed resource. It also initiates immediate check actions on other resource groups that are dependent on this resource group.

If the resource restarts successfully, the resource group transitions to the ONLINE state and Oracle Clusterware performs pullup dependency evaluation within and across resource groups.

If Oracle Clusterware exhausts all local restart attempts of the resource, it stops the entire resource group. It also immediately stops other resource groups with a stop dependency on the resource group. Oracle Clusterware attempts a local restart of the resource group, if configured to do so. On exhausting all restart attempts, the resource group will fail over to another server in the cluster.

Failure and Recovery of Non-Critical Resources

1. If a non-critical resource in a resource group fails, Clusterware attempts a local restart of the failed resource according to the `RESTART_ATTEMPTS` and `UPTIME_THRESHOLD` values.
 - There is no impact on the state of the resource group when a non-critical resource fails.
2. If the resource restarts successfully, Clusterware performs pullup dependency evaluation and corresponding startup actions.
3. If Clusterware exhausts all local restart attempts of the resource, there is no impact on the state of the resource group.
 - You must explicitly start the non-critical resource after fixing the cause of failure.

ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

When a non-critical resource in a resource group fails, Oracle Clusterware attempts a local restart of the failed resource according to the values of the `RESTART_ATTEMPTS` and `UPTIME_THRESHOLD` resource attributes. There is no impact on the state of the resource group when a non-critical resource fails.

Oracle Clusterware initiates immediate check actions on other resources in the same group that have a stop dependency on the failed resource. If the resource restarts successfully, Oracle Clusterware performs pullup dependency evaluation and corresponding startup actions. If Oracle Clusterware exhausts all local restart attempts of the resource, there is no impact on the state of the resource group. You must then explicitly start the non-critical resource after fixing the cause of failure.

Resource Group Types

- In Oracle Clusterware, a resource type is a template for a class of resources.
- Similarly, resource group types provide a commonly applicable set of attributes to all resource groups.
- When you create a resource group, you must specify a resource group type.
- Oracle Clusterware provides two base resource group types:
 - Use the `local_resourcegroup` type to create a resource group that contains only local resources.
 - The `cluster_resourcegroup` type can have one or more instances running on a static or dynamic set of servers in the cluster.



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

In Oracle Clusterware, a resource type is a template for a class of resources. Resource group types provide a commonly applicable set of attributes to all resource groups. When you create a resource group, you must specify a resource group type.

Oracle Clusterware provides two base resource group types: `local_resourcegroup` and `cluster_resourcegroup`. The base resource types have attributes that are similar to resources, some of which you can configure.

Local Resource Group Type

Use the `local_resourcegroup` type to create a resource group that contains only local resources.

Instances of a resource group of this type can run on each node in the cluster. For a comprehensive list of local resource group type attributes, refer to the *Oracle Clusterware Administration and Deployment Guide 12c Release 2 (12.2)*.

Cluster Resource Group Type

A resource group of type `cluster_resourcegroup` can have one or more instances running on a static or dynamic set of servers in the cluster. Such a resource group can fail over to another server in the cluster according to the placement policy of the group. For a comprehensive list of cluster resource group type attributes, refer to the *Oracle Clusterware Administration and Deployment Guide 12c Release 2 (12.2)*.

Using Resource Groups

Use `crsctl` to create resource groups, resource group types, and add resources to resource groups.

1. Use the following command to create a resource group:

```
$ crsctl add resourcegroup rg1 -type local_resourcegroup
```

2. To create a resource group based on a custom resource group type, you must first create the resource group type.

```
$ crsctl add resourcegroup-type rgt1 -basetype local_resourcegroup
```

3. After you create a resource group, you can begin to add resources to the resource group.

```
$ crsctl add resource my_apache -group rg1
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Use `crsctl` to create resource groups, resource group types, and add resources to resource groups. To use resource groups, you must first create the resource group based on either a built-in resource group type or a resource group type that you create. After you have created a resource group, you can add resources to it.

1. Use the following command to create a resource group:

```
$ crsctl add resourcegroup group_name -type group_type
```

This command creates an empty resource group into which you can add resources. You must provide a name for the resource group and a resource group type. If you choose to base your resource group on a custom resource group type, you must first create the resource group type.

2. If you want to create a resource group based on a custom resource group type, you must create the resource group type as follows:

```
$ crsctl add resourcegroup-type group_type_name -basetype \
base_group_type {-file file_path | -attr \
"attribute_name=attribute_value"}
```

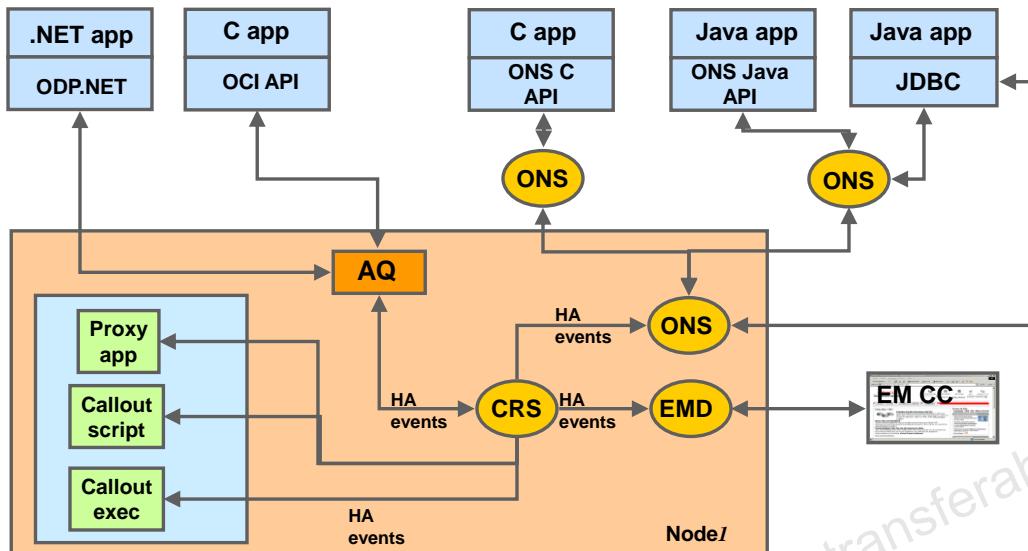
The preceding command creates a resource group type that provides a singular set of attributes for any resource group that you created based on this resource group type. You must provide an existing resource group type as a base resource group type, and either a path to a file that contains a line-delimited list of attribute/attribute value pairs or alternatively, you can provide a comma-delimited list of attribute/attribute value pairs on the command line.

3. After you create a resource group, you can begin to add resources to the resource group as follows:

```
$ crsctl add resource resource_name -group group_name
```

The resource group to which you add a resource must exist and the resource that you are adding must be in offline state. A resource can be a member of only one resource group. If you have a resource that is shared by multiple applications, such as a file system, it is recommended that you put these resources into their own individual resource groups.

HA Events: ONS and FAN



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

HA events are generated when resources change state within an Oracle Clusterware environment. Oracle Notification Service (ONS) is a facility that creates a bridge with middle-tier servers or applications to transport these events to application logic for handling or reaction. ONS is part of a larger framework known as Fast Application Notification (FAN). With FAN, applications use these events to achieve very fast detection of failures and rebalancing of connection pools, following failures and recovery. When FAN is used together with an Oracle Database, the Advanced Queuing (AQ) feature allows HA events to be received by external applications such as .NET clients. The easiest way to receive all the benefits of FAN, with no effort, is to use a client that is integrated with FAN, such as:

- Java Database Connectivity (JDBC) Implicit Connection Cache
- User-extensible callouts
- Connection Manager (CMAN)
- Listeners
- Oracle Notification Service (ONS) API
- OCI Connection Pool or Session Pool
- Transparent Application Failover (TAF)
- ODP.NET Connection Pool

Note: Not all the preceding applications can receive all types of FAN events.

Managing Oracle Notification Server with `srvctl`

- To determine the current ONS configuration:

```
$ srvctl config nodeapps -s
ONS exists: Local port 6100, remote port 6200, EM port 2016, Uses SSL false
ONS is enabled
ONS is individually enabled on nodes:
ONS is individually disabled on nodes:
```

- To add Oracle Notification Server:

```
$ srvctl add nodeapps -onslocalport 6100 -onsremoteport 6200
```

- To start or stop Oracle Notification Server:

```
$ srvctl start|stop nodeapps
```



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

The `srvctl` utility can be used to create, manage, and remove the Oracle Notification Server (ONS). To add an ONS, run the following command:

```
srvctl add nodeapps [-l ons_local_port] [-r ons_remote_port]
[-t host[:port], [host[:port]], ...]
```

where `-l` identifies the ONS daemon local (or listener) port, `-r` indicates the ONS daemon remote (or write) port, and `-t` specifies a list of `host:port` pairs of remote hosts that are part of the ONS network but are not part of the Oracle Clusterware cluster. The local port is used for communication between the ONS process and the ONS clients on the same node. A remote port is defined in the OCR that is used for communication between the ONS process and other ONS processes on other cluster nodes or middle-tier nodes.

The syntax to start and stop the ONS is as follows:

```
srvctl start|stop nodeapps [-f]
```

To display configuration information for ONS (and all other `nodeapps`), run the following command:

```
srvctl config nodeapps
```

To get the environment variables for ONS (and all other `nodeapps`), run the following command:

```
srvctl getenv nodeapps
```



Quiz

When Oracle Clusterware is installed, the **GENERIC** and **FREE** server pools are created automatically.

- a. True
- b. False



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.



Quiz

Oracle Database 12c offers two methods for managing resources. These methods are:

- a. Administration-based management
- b. Threshold-based management
- c. Policy-based management



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.



Quiz

The crsctl add resourcegroup command is used to add Clusterware resource groups.

- a. True
- b. False



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Describe the high-availability components of Oracle Clusterware
- Contrast policy-managed and administration-managed databases
- Create an application Virtual IP (VIP)
- Manage application resources
- Describe and implement Clusterware resource groups



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Practice 11: Overview

This practice covers the following topics:

- Configuring highly available application resources on flex cluster leaf nodes
- Creating Clusterware Resource Groups



Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Unauthorized reproduction or distribution prohibited. Copyright© 2019, Oracle and/or its affiliates.

GANG LIU (gangl@baylorhealth.edu) has a non-transferable license
to use this Student Guide.