ORACLE UNIVERSITY

Hardware and Software
Engineered to Work Together

Oracle Database 12*c*: Clusterware Administration

Student Guide
D81246GC10
Edition 1.0 | September 2014 | D85450

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

ORACLE

Author

Jim Womack

Technical Contributors and Reviewers

Allan Graves

Gerlinde Frenzen

Branislav Valny

Herbert Bradbury

Ira Singer

Harald Van Breederode

Joel Goodman

Sean Kim

Andy Fortunak

Al Flournoy

Markus Michalewicz

Maria Billings

Editor

Raj Kumar

Graphic Designer

Seema Bopaiah

Publishers

Jobi Varghese

Joseph Fernandez

Jayanthy Keshavamurthy

Contents

1

Introduction to Clusterware

ORACLE

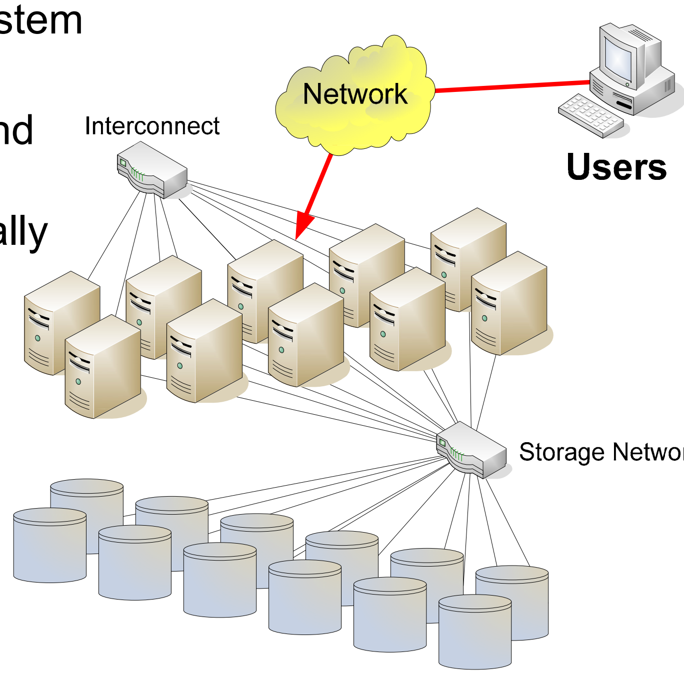
Objectives

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

What Is a 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



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.

What Is Clusterware?

- Clusterware is 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

ORACLE

Clusterware is a term used to describe software that provides interfaces and services that enable and support a cluster.

Different cluster architectures require clusterware that delivers different services. For example, in a simple failover cluster, the clusterware may monitor the availability of applications and perform a failover operation if a cluster node becomes unavailable. In a load balancing cluster, different services are required to support workload concurrency and coordination.

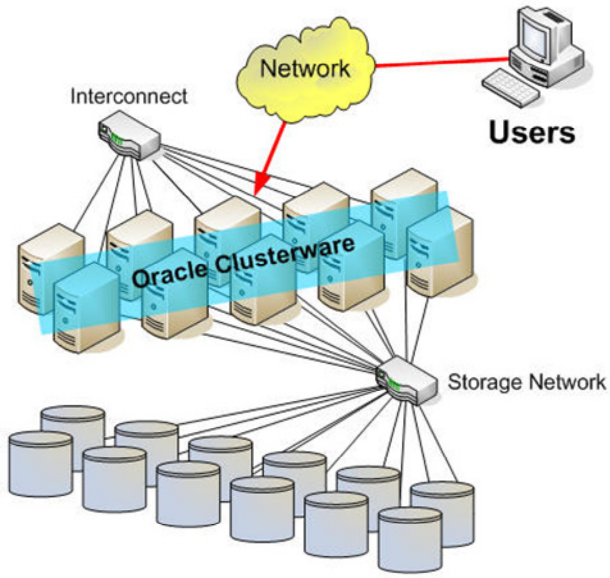
Typically, clusterware includes capabilities that:

- Allow the cluster to be managed as a single entity (not including OS requirements), if desired
- Protect the integrity of the cluster so that data is protected and the cluster continues to function even if communication with a cluster node is severed
- Maintain a registry of resources so that their location is known across the cluster and so that dependencies between resources is maintained
- Deal with changes to the cluster such as node additions, removals, or failures
- Provide a common view of resources such as network addresses and files in a file system

Oracle Clusterware

Oracle Clusterware is:

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



ORACLE

Oracle Clusterware is a key part of Oracle Grid Infrastructure, which also includes Automatic Storage Management (ASM) and the Oracle Cloud 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.

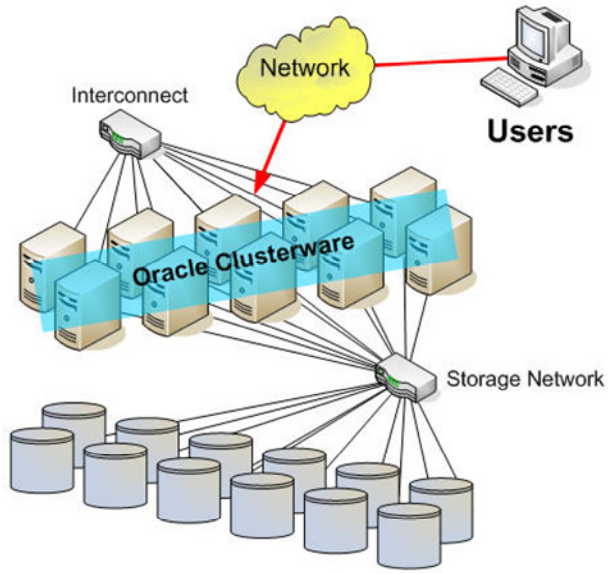
The combination of Oracle Clusterware, ASM, and ASM Cluster File System (ACFS) provides administrators with a unified cluster solution that is not only the foundation for the RAC database, but can also be applied to all kinds of other applications. Oracle Clusterware also manages resources, such as virtual 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 12*c* 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)

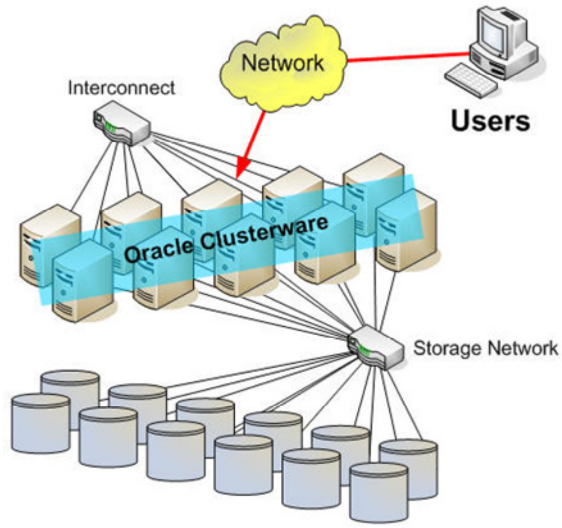


Oracle Clusterware provides a complete set of cluster services to support the shared disk, load balancing cluster architecture of the Oracle Real Application Cluster (RAC) database. Oracle Clusterware can also be used to provide failover clustering services for single-instance Oracle databases and other 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 Clusterware has become the required clusterware for Oracle Real Application Clusters (RAC). Oracle Database 12*c* 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 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 Management Platform Interface).

ORACLE

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 11*g* 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 Management Platform 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 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.

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.

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.

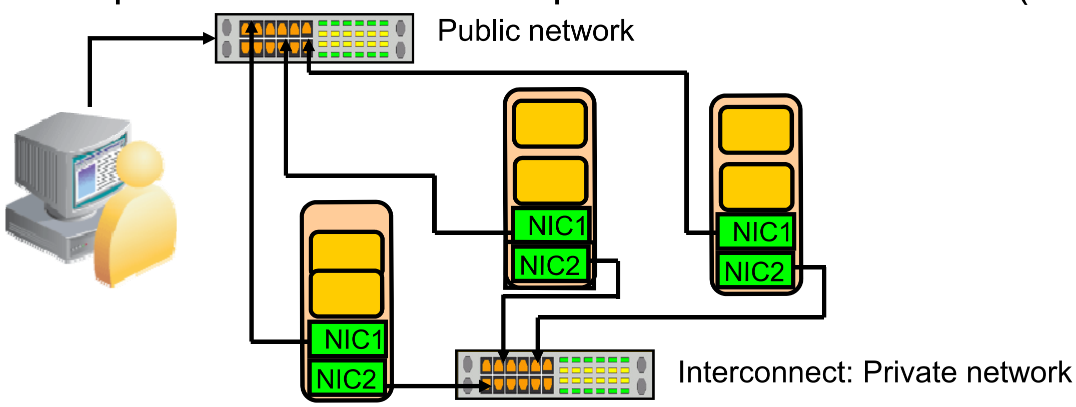
Each server must have an operating system that is certified with the Oracle Clusterware version you are installing. Refer to the certification matrices available in the Oracle Grid Infrastructure Installation Guide for your platform or on My Oracle Support (formerly Oracle MetaLink) for details, which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



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 make use of 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.

Oracle Clusterware 12*c* 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 addresses or IPv4 addresses. After installation, you can add IPv4 addresses to an IPv6 cluster or IPv6 addresses to an IPv4 cluster.

Note: You must configure the private network as an IPv4 network. IPv6 addresses are not supported for the interconnect.

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

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.

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

In previous releases, the GNS was dedicated to one Oracle Grid Infrastructure–based cluster, providing name resolution only for its own cluster member nodes. With the release of Oracle Grid Infrastructure 12*c*, one Oracle GNS can now manage just the cluster member nodes in its own cluster, or GNS can provide naming resolution for all nodes across all clusters in the data center that are delegated to Oracle GNS for resolution.

Using only one GNS for all nodes that are part of an Oracle Grid Infrastructure cluster in the data center not only streamlines the naming convention, but also enables a data center cloud, minimizing day-to-day administration efforts.

Grid Naming Service Configuration Options

- GNS can run in either automatic or standard cluster 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.

ORACLE

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 CloudFS features that require name resolution information within the cluster. However, any node additions or changes must be carried out as manual administration tasks.

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.

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

Answer: c

Statement c is not correct. The GNS VIP address must be statically defined.

Quiz

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

a. True
b. False

Answer: b

This statement is false. Actually, each cluster node's public Ethernet adapter should support TCP/IP. The private adapter should support UDP or RDS on Linux/UNIX platforms and TCP on Windows platforms.

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

2

Oracle Clusterware Architecture

Objectives

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

- Architecture
- Startup details

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

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

ORACLE

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 11*g* 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

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, cssdagent`	`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 (r)`	`orarootagent.exe`
Sysmon	`osysmond.bin (r)`	`osysmond.exe`

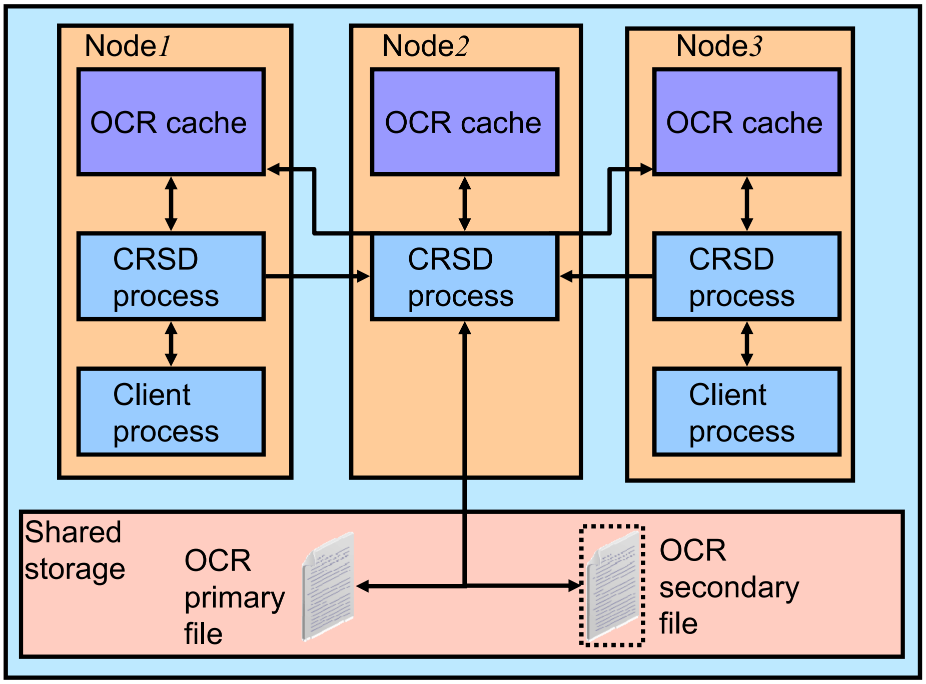
ORACLE

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 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)

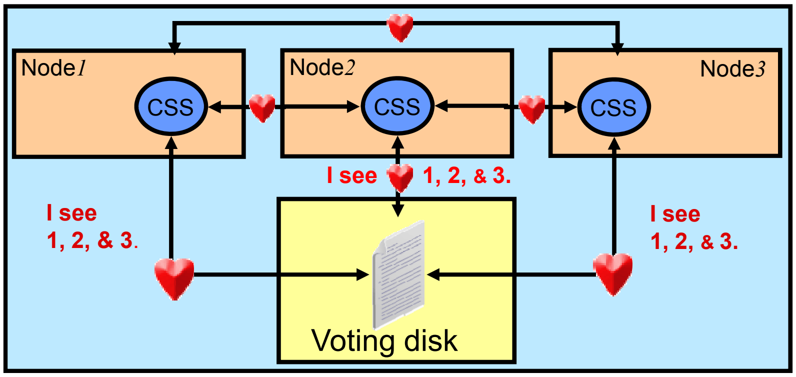


Cluster configuration information is maintained in the OCR. You can have up to five OCR locations. Each OCR location must reside on shared storage that is accessible by all of the hub nodes in the cluster. The OCR relies on distributed shared cache architecture for optimizing queries, and clusterwide 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.

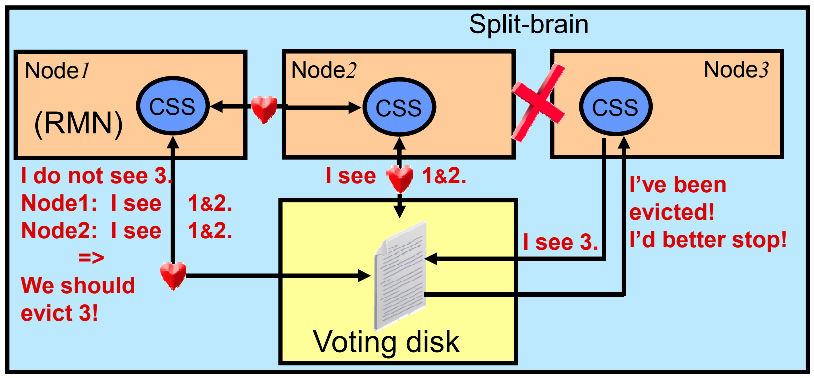
Note: In the diagram in the slide, note that a client process might also exist on node 2 but is not shown for the sake of clarity.

CSS Voting Disk Function



Nodes can see each other.

Node3 can no longer communicate through private interconnect. Others no longer see its heartbeats and evict that node by using the voting disk.



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
- If you do not configure voting disks on Oracle ASM:
  - For high availability, create a minimum of three voting disks in separate locations, avoiding a single point of failure
  - Use external mirroring to provide redundancy if you configure a single voting disk
- Do not use more than five voting disks, even though a maximum of 15 voting disks is supported.

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.

If you do not configure voting disks on Oracle ASM, then for high availability, Oracle recommends that you have a minimum of three voting disks on physically separate storage. This avoids having a single point of failure. If you configure a single voting disk, you must use external mirroring to provide redundancy. Although Oracle Clusterware supports up to 15 redundant voting disks, it is recommended to use no more than 5.

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`

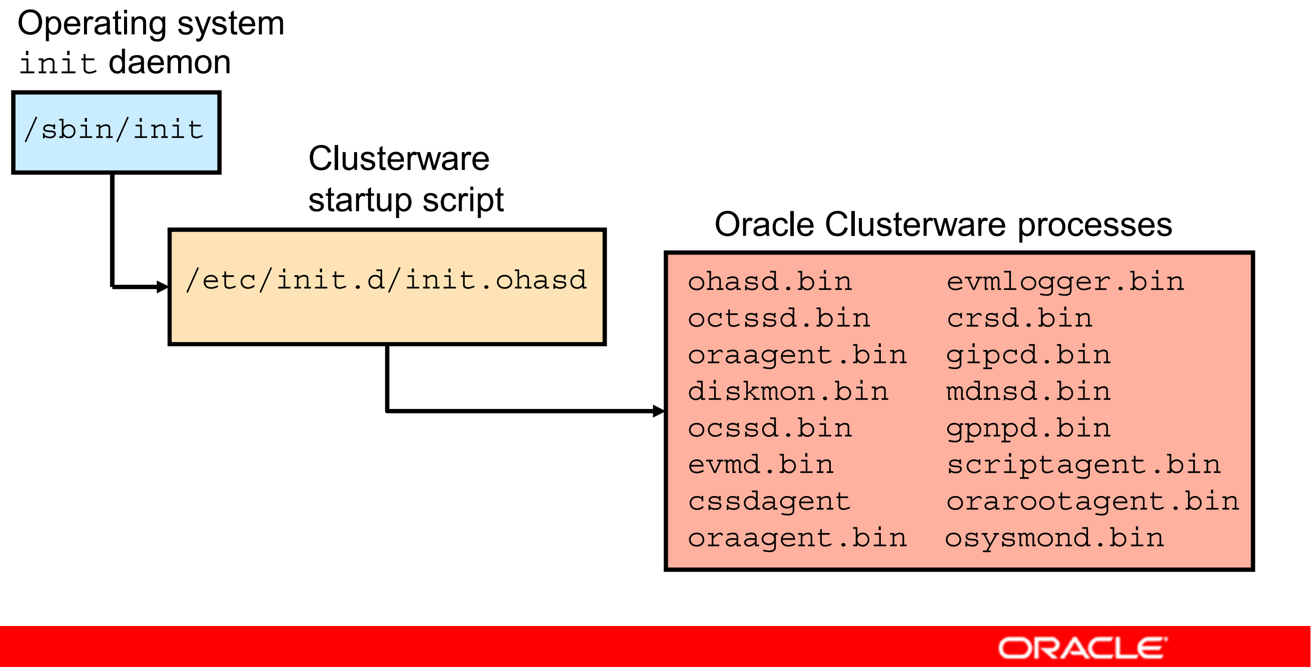
In Oracle Clusterware 12*c*, each node in a cluster has a local registry for node-specific resources, called an Oracle Local Registry (OLR). The OLR is a registry similar to the OCR, but contains information specific to each node. It contains manageability information about Oracle Clusterware, including 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.



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 when the OS `init` daemon calls the `/etc/init.d/init.ohasd` startup script.
- On OL5, Oracle Clusterware installation modifies `/etc/inittab` to restart `ohasd` in the event of a crash.

```
# cat /etc/inittab
..
h1:35:respawn:/etc/init.d/init.ohasd run >/dev/null 2>&1 </dev/null
```

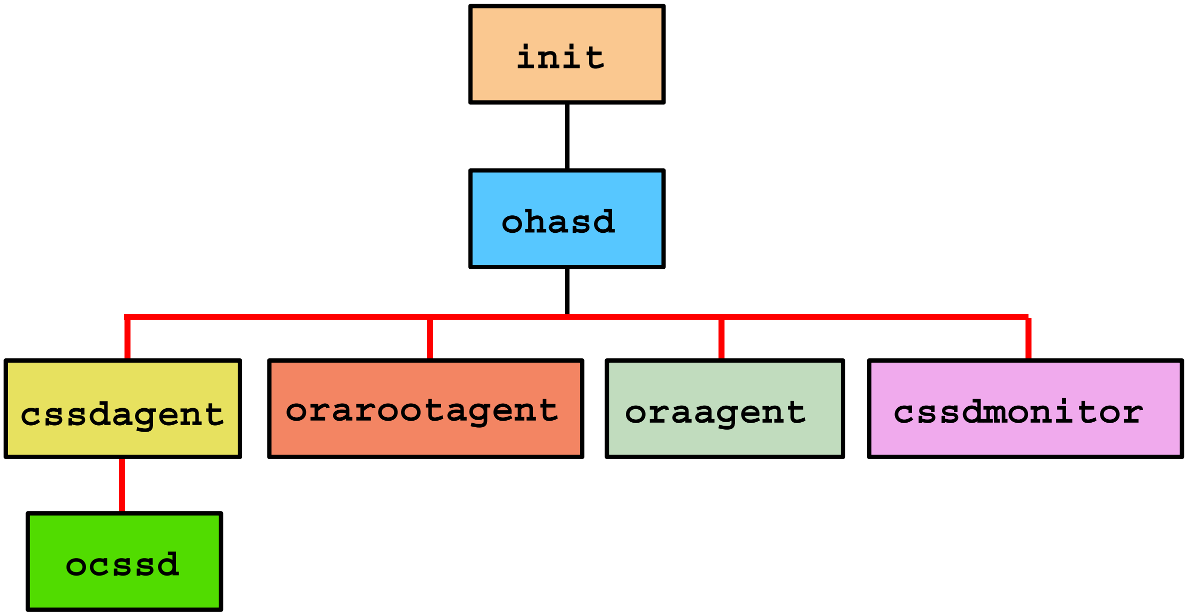
- On OL6, Clusterware startup is controlled by Upstart via the `/etc/init/oracle-ohasd.conf` file.

```
# 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 OL5 systems, Clusterware startup and crash behavior is handled by `init` based on an entry in the `/etc/inittab` file. On OL6 systems, Clusterware startup and crash behavior is still handled by `init`, but the `/etc/inittab` only defines the default runlevel for that node. Service startup is controlled by the Upstart boot service based on service definition files located in the `/etc/init` directory. The primary advantage of Upstart over the traditional System V init method is boot speed. Unlike System V init, which starts services serially, Upstart is event driven. Services are started only when they are needed.

Clusterware Startup Details



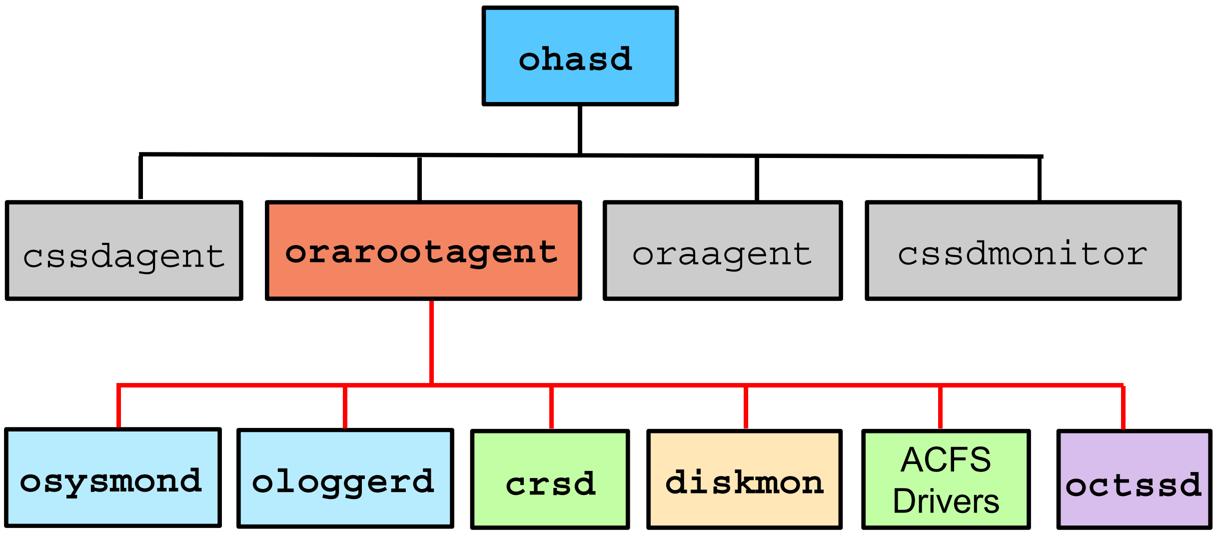
The Oracle High Availability Services daemon (`ohasd`) is responsible for starting in proper order, monitoring, and restarting other local Oracle Clusterware daemons, up through the `crsd` daemon, which in turn manages clusterwide resources.

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 11*g* Release 1 (11.1).
- The cssdmonitor is started and is responsible for monitoring the ocssd daemon.

Clusterware Startup: OHASD `orarootagent`

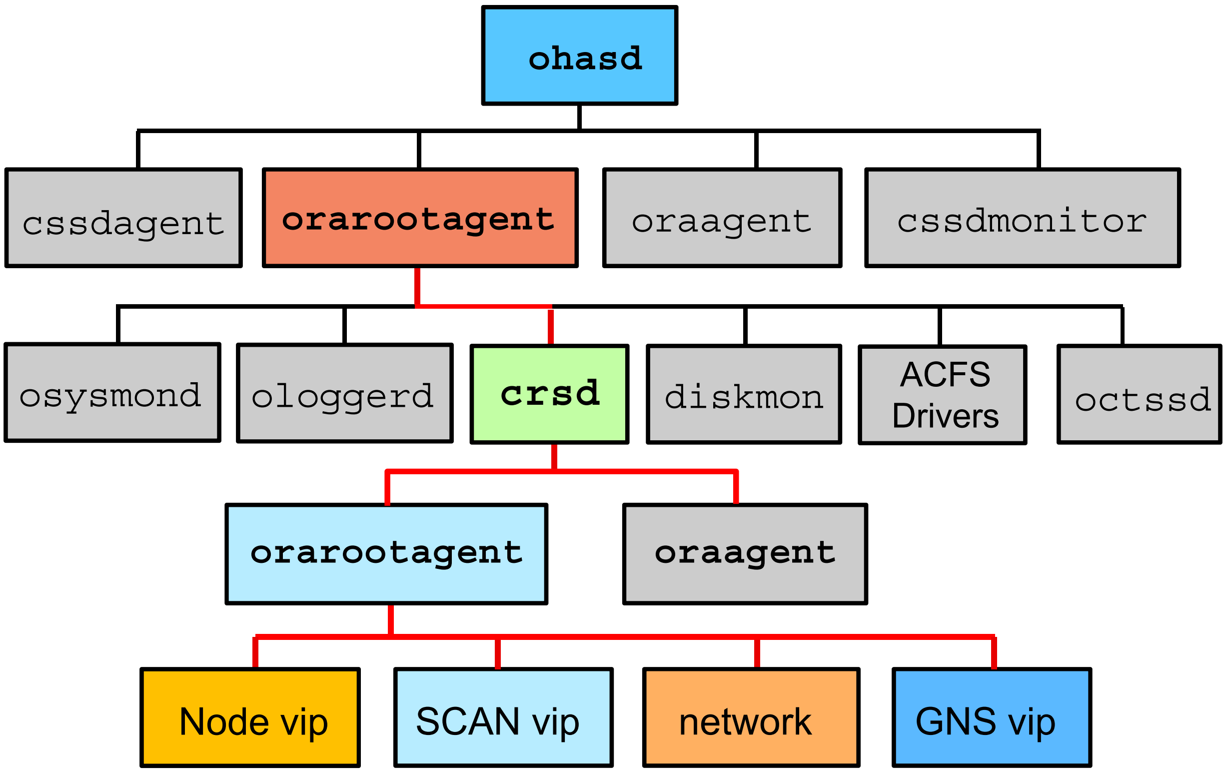


The `orarootagent` process is responsible for starting the following processes:

- **`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 a 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.

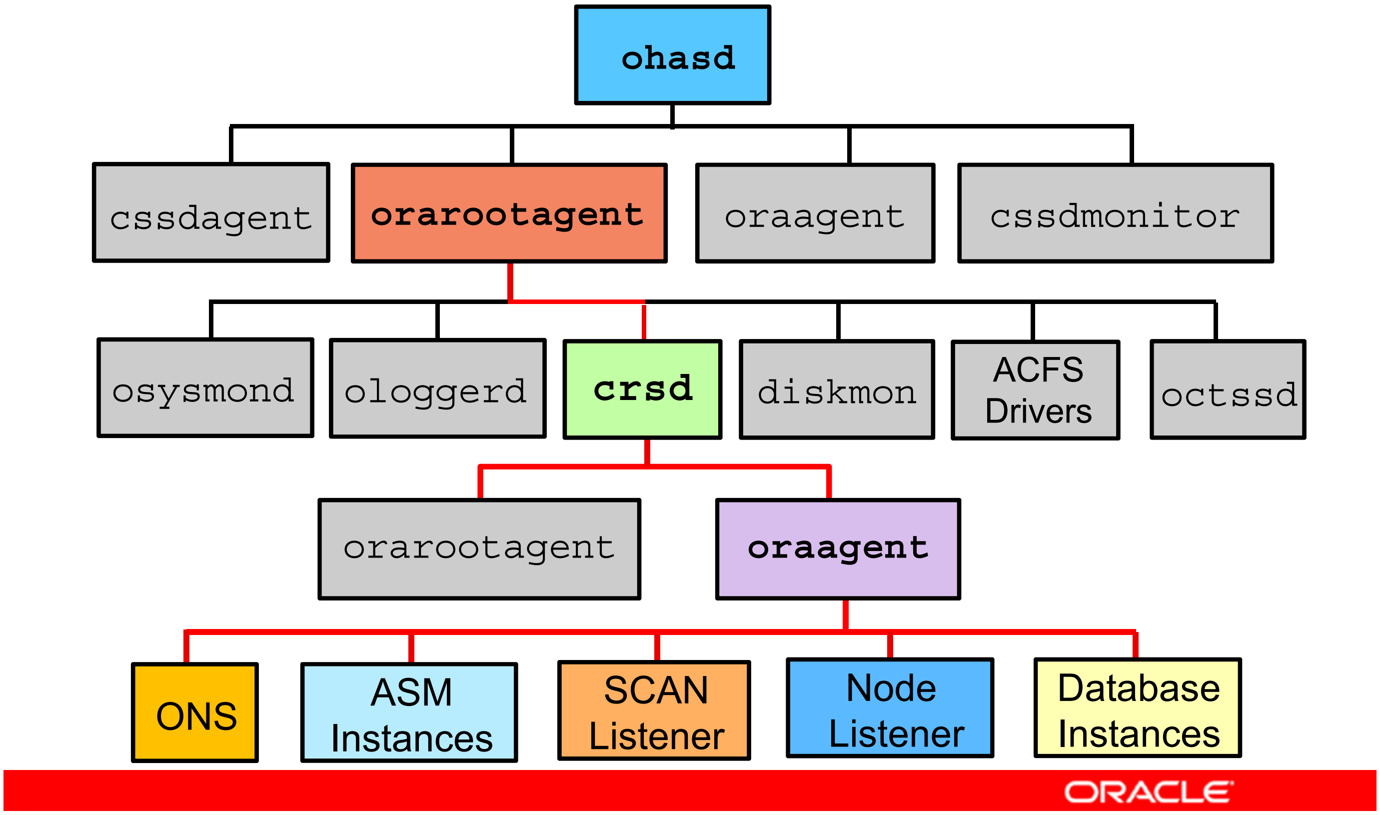
Clusterware Startup Details: CRSD `orarootagent`



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.

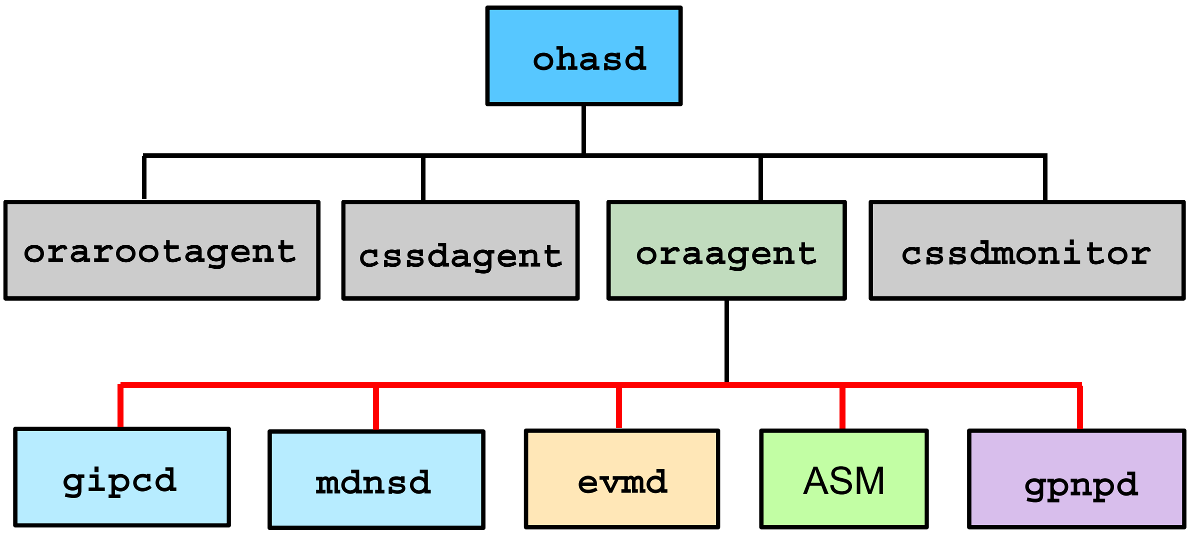
Clusterware Startup Details: CRSD `oraagent`



As noted in the previous slide, the `crsd` process starts another `orarootagent` process and another `oraagent` process. The new `oraagent` process is responsible for starting the following resources:

- **ONS:** The ONS or Oracle Notification Service is a publishing and subscribing service for communicating Fast Application Notification (FAN) events.
- **ASM Instances:** The Oracle ASM instances provide disk management for Oracle Clusterware and Oracle Database. One ASM instance is started on each cluster node.
- **SCAN Listener:** Three SCAN Listeners are started on the cluster nodes where the SCAN VIPs are started. Oracle Database 11*g* Release 2 and later instances register with SCAN listeners only as remote listeners.
- **Node Listener:** 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.
- **Database Instances:** If the cluster nodes are supporting an Oracle RAC database, the database instances are started.

Clusterware Startup Details: OHASD `oraagent`



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`:** Grid Plug and Play (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 nodes:

```
# crsctl start cluster -all
```

- 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
```


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 `SLCS_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
**************************************************************
```


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.

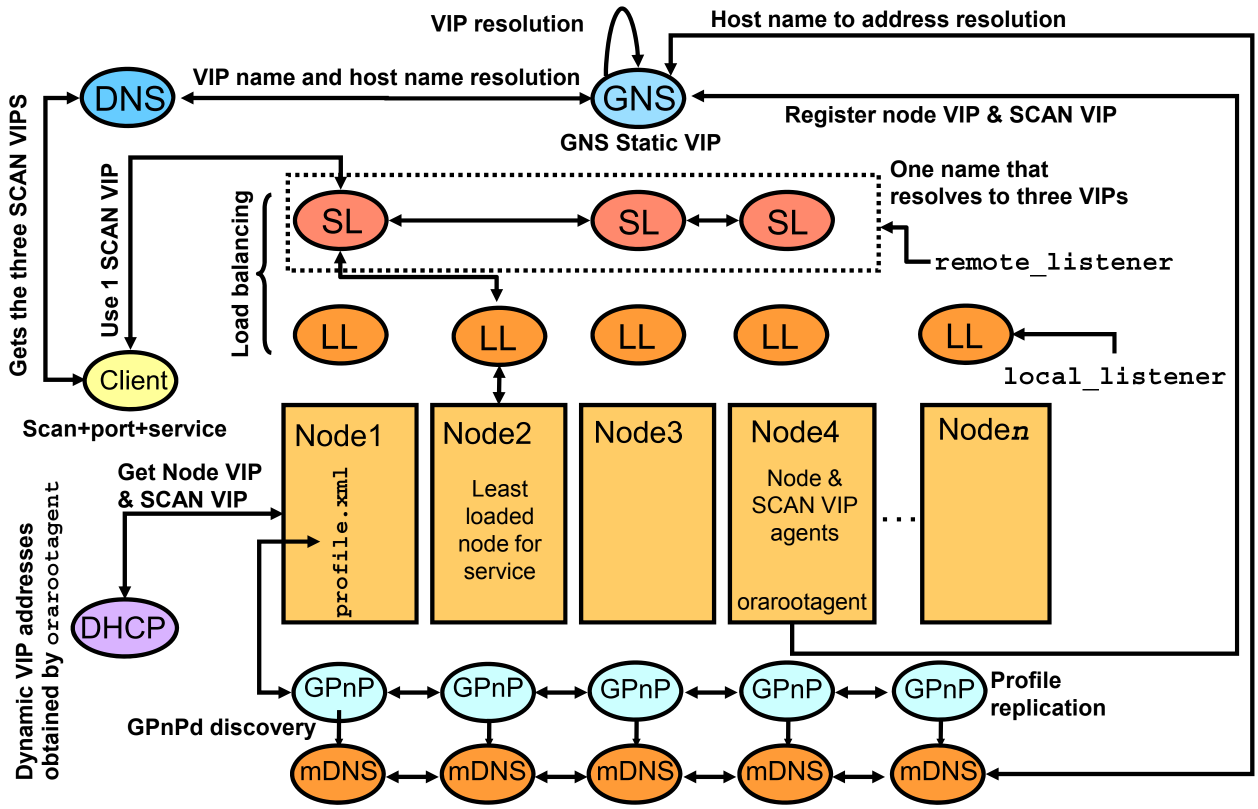
Viewing the High Availability Services Stack

```
$ 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
...
```


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

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

GPnP 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.

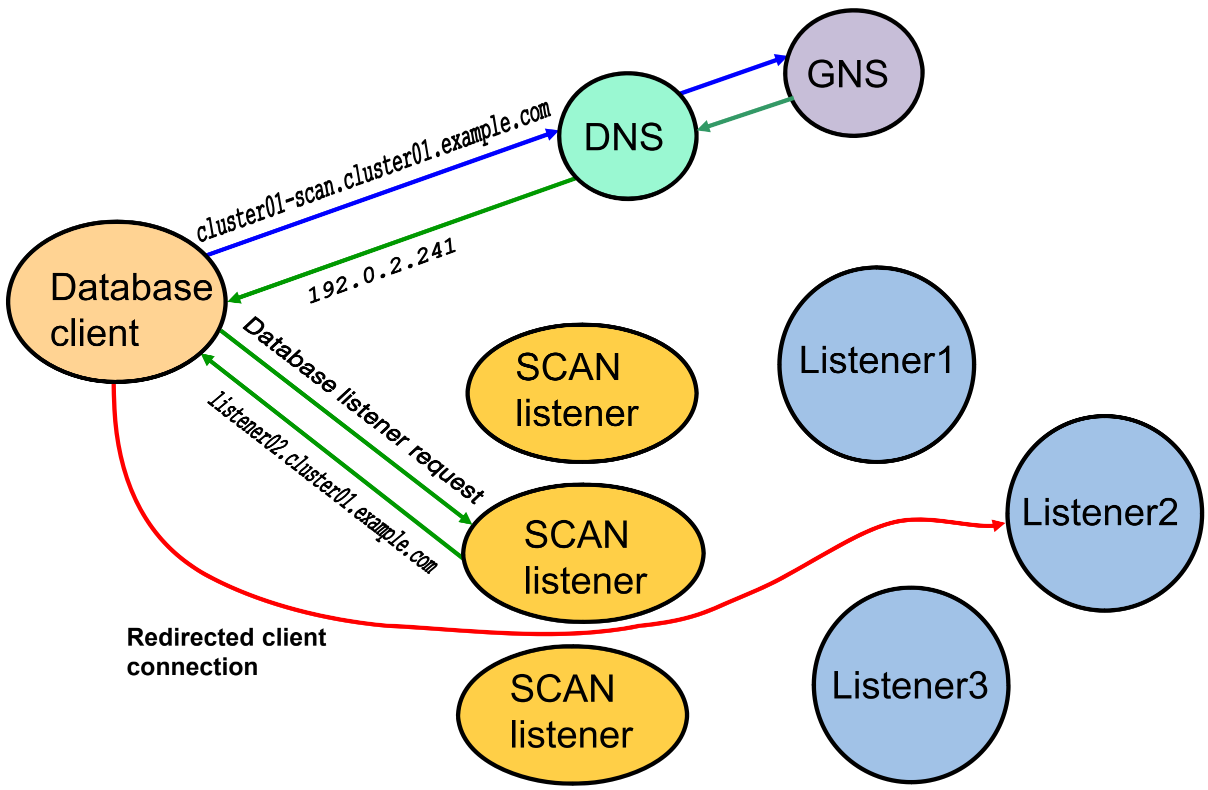
How GPnP Works: Cluster Node Startup

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

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



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 5 or RHEL 5, the `init.ohasd` entry in the `/etc/inittab` 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

Answer: d

The answer is d. The `init.ohasd` entry in the `/etc/inittab` file is responsible for restarting the Oracle high availability daemon (`ohasd`) in the event of a crash.

Summary

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

- Architecture
- Startup details

3

Flex Clusters

Objectives

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

- Flex Cluster architecture and components
- The effect of node failure in a Flex Cluster

Flex Clusters: Overview

- In previous releases, most large clusters contain between 32 and 64 nodes.
- With Oracle Clusterware 12*c*, Flex Clusters are designed to scale well up to 2000 nodes. Use cases include:
  - Large pools of highly available application resources
  - Multiple databases and applications running in one cluster

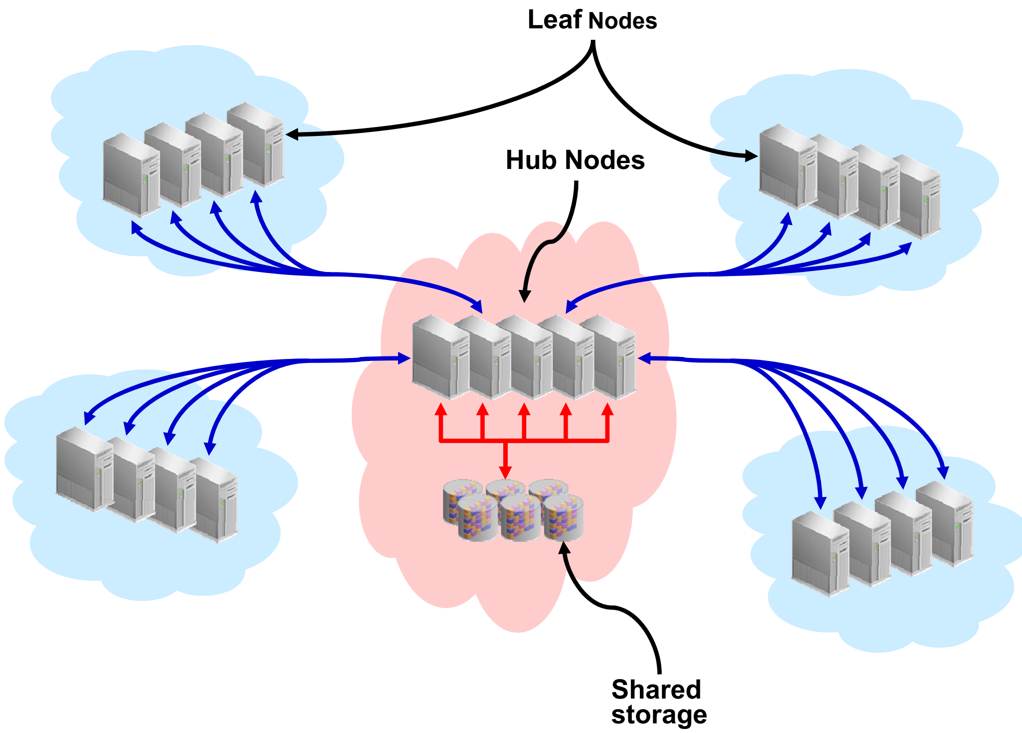
Previous releases of Oracle Clusterware have been used to build large production clusters containing between 32 and 64 nodes. A few clusters larger than 100 nodes have been successfully deployed.

With Oracle Clusterware 12*c*, a new set of features enables Flex Clusters. In this release, Flex Clusters are designed to scale well up to 2000 nodes.

In release 12.1, you can use Flex Clusters to:

- Manage large pools of application resources with high-availability and failover protection
- Efficiently support multiple highly available databases and applications running in a single cluster

Flex Cluster Architecture



Flex Clusters use a hub-and-spoke topology, as illustrated in the slide.

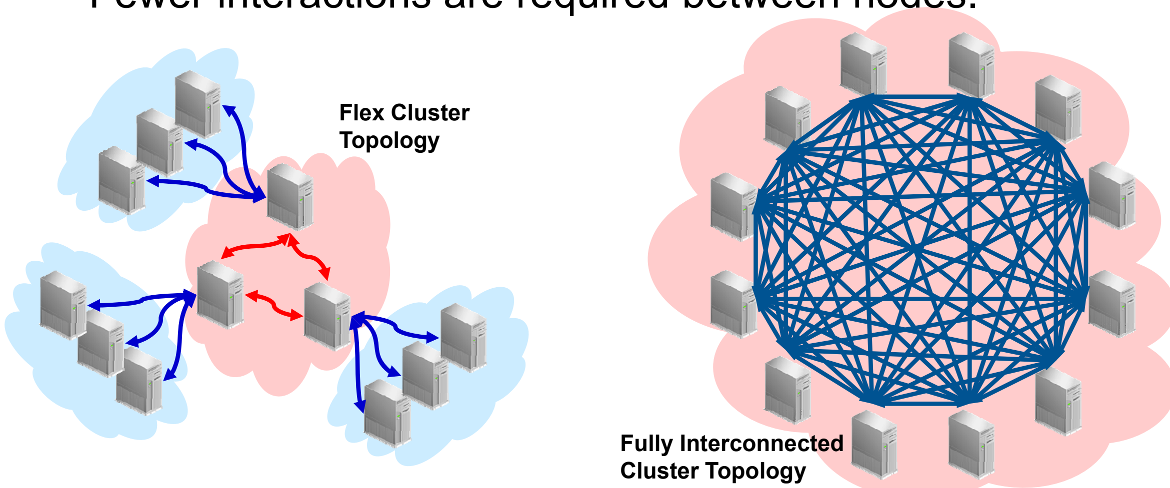
The core of a Flex Cluster is a group of Hub Nodes. The group is essentially the same as a release 11.2 cluster, and can scale up to the size of an existing release 11.2 cluster. There must be one, and only one, group of Hub Nodes in a Flex Cluster deployment, and like a release 11.2 cluster, each Hub Node must be connected to storage that is shared across the group of Hub Nodes.

Zero or more Leaf Nodes may be connected to a Flex Cluster. Each Leaf Node is connected to the cluster through a Hub Node. Leaf Nodes do not require direct access to the shared storage connected to the Hub Nodes.

Flex Cluster Scalability

The Flex Cluster hub-and-spoke topology segments the cluster into more manageable groups of nodes.

- Only the Hub Nodes require direct access to the OCR and voting disks.
- Fewer interactions are required between nodes.



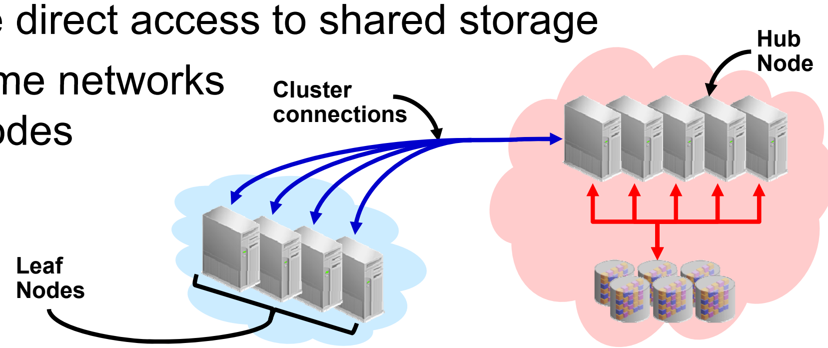
The two-layered hub-and-spoke topology is the key architectural feature that allows a Flex Cluster to scale well beyond previous limits. In essence, the hub-and-spoke topology segments the cluster into groups of nodes, and each group contains a manageable number of nodes. This segmentation has two fundamental effects:

- By limiting the size of the hub, contention for key clusterware resources, such as the Oracle Cluster Registry (OCR) and voting disks, does not increase significantly due to the addition of the Leaf Nodes. This is important because contention for the voting disks can lead to nodes being evicted from a cluster.
- Fewer network interactions are required between nodes in the cluster. Consequently, there is less administrative network traffic, such as heartbeats, exchanged between the nodes. This is illustrated in the diagram in the slide. On the left side, the 12-node Flex Cluster contains 12 interaction paths. On the right side, the fully interconnected 12-node cluster contains 66 possible interaction paths. For a 1000-node cluster, the difference would be far more noticeable. Assuming 40 Hub Nodes, with 24 Leaf Nodes per Hub Node, a Flex Cluster contains 1740 possible interaction paths. In comparison, a 1000-node fully interconnected cluster contains 499500 interaction paths.

Leaf Node Characteristics

Leaf Nodes:

- Are more loosely coupled to a cluster than are Hub Nodes
- Automatically discover the Hub Nodes at startup
- Connect to the cluster through a Hub Node
  - Failure of the Hub Node or network failure results in eviction of associated Leaf Nodes.
  - Functioning Leaf Nodes can be brought back into the cluster.
- Do not require direct access to shared storage
- Are on the same networks as the Hub Nodes



Compared to Hub Nodes, Leaf Nodes are loosely coupled to the cluster. When Oracle Clusterware is started on a Leaf Node, the Leaf Node automatically discovers the Hub Nodes and is associated with a single Hub Node. For cluster membership purposes, Hub Nodes periodically exchange heartbeat messages with their associated Leaf Nodes. Similar mechanisms are used for other services.

If a Hub Node fails, or if the network link between a Hub Node and a Leaf Node fails, the associated Leaf Nodes may be removed from the cluster. In any case, if there is no fault with the Leaf Node, it can be brought back into the cluster by restarting Oracle Clusterware on the Leaf Node.

A Leaf Node does not require direct access to shared storage. This means that Leaf Nodes can participate in the cluster without storage-related hardware and network connections, such as fiber-channel network connections and host bus adapters.

In release 12.1, all Leaf Nodes are on the same public and private networks as the Hub Nodes.

Grid Naming Service (GNS) and Flex Clusters

Clients on Leaf Nodes use GNS to locate Hub Node services.

- The GNS server location is stored in the cluster profile.
- Leaf Node services issue DNS queries to GNS.
  - Particularly during Leaf Node startup
- A fixed GNS VIP is required on one of the Hub Nodes.
  - The reason is that Leaf Node clients need a reliable, well-known location to contact.
- DNS forwarding is not required for Flex Clusters.
  - But it can be implemented to better integrate GNS with DNS.

Flex Clusters require the Grid Naming Service (GNS) to be configured with a fixed virtual IP (VIP) on one of the Hub Nodes.

GNS is used to dynamically register and resolve names within the cluster. In particular, GNS is referenced because the Leaf Node is associated with a Hub Node for cluster membership purposes during the clusterware start-up process on a Leaf Node. This requires access to GNS through a fixed VIP running on one of the Hub Nodes, so that Leaf Nodes have a reliable, well-known naming service within the cluster.

Domain Name Server (DNS) forwarding is not required to facilitate discovery of clusterware services by Leaf Nodes; however, it can still be configured to integrate GNS with a wider network-naming service.

Cluster Mode: Overview

- Oracle Clusterware 12*c* introduces a new cluster mode setting to enable the Flex Cluster functionality.
  - Users must explicitly enable Flex Cluster functionality.
- The default cluster mode setting disables Flex Cluster functionality.
  - Users who do not implement Flex Clusters are not exposed to the new code.
  - The performance and stability of standard clusters are not affected by Flex Cluster functionality.

You can choose to enable or disable Flex Cluster functionality by using the new cluster mode setting. By default, Flex Cluster functionality is disabled.

Configuring the Cluster Mode

- Showing the current cluster mode:

```
$ crsctl get cluster mode status
```

- Converting from a standard cluster to a Flex Cluster:
  1. Ensure that GNS is configured with a fixed VIP:

```
# srvctl add gns -i <GNS_VIP_address> -d <cluster_domain>
```

  2. Enable Flex ASM in the cluster using ASMCA.
  3. Set the cluster mode:

```
# crsctl set cluster mode flex
```

  4. Stop Oracle Clusterware on each node:

```
# crsctl stop crs
```

  5. Start Oracle Clusterware on each node:

```
# crsctl start crs
```


At any time, a cluster administrator or system administrator can check the current cluster mode by using the `crsctl get cluster mode status` command.

To convert from a standard cluster to a Flex Cluster, an administrator should first ensure that GNS is configured with a fixed VIP. If GNS is not configured with a fixed VIP, the remainder of the procedure will fail. Also, Flex ASM must be enabled on the cluster before setting the cluster mode. Next, the system administrator (`root`) can set the cluster mode by using the `crsctl set cluster mode flex` command. Finally, the system administrator must restart the cluster by using the `crsctl stop crs` command on each node in the cluster followed by the `crsctl start crs` command on each node in the cluster. Note that you cannot avoid cluster downtime when changing the cluster mode.

To convert from a Flex Cluster to a standard cluster, the system administrator must set the cluster mode by using the `crsctl set cluster mode standard` command and clusterware must be stopped and restarted across the cluster. There is no requirement to reconfigure GNS or Flex ASM, because the configuration required for Flex Cluster mode is also compatible with standard cluster mode.

Note that any node that is unable to join the reconfigured cluster is left out of the cluster and eventually dropped from it. This can occur when, for example, a Leaf Node having no access to shared storage cannot join a cluster converted to standard mode.

Configuring the Node Role

- Showing the current node role:

```
$ crsctl get node role status -node <hostname>
```

```
$ crsctl get node role status -node host02
Node 'host02' active role is 'hub'
```

- Setting the node role:

```
# crsctl set node role [ hub | leaf | auto ] -node <hostname>
```

```
# crsctl set node role leaf -node host02
# crsctl get node role config -node host02
Node 'host02' configured role is 'leaf'
# crsctl get node role status -node host02
Node 'host02' active role is 'hub'
```

ORACLE

One of the key configuration tasks for a Flex Cluster is specifying which nodes are Hub Nodes and which nodes are Leaf Nodes.

At any time, a cluster administrator or system administrator can check the current role for a node by using the following command:

`crsctl get node role status -node <hostname>`

Configuring the node role can be achieved in two ways:

- A system administrator can explicitly specify the node role as `hub` for a Hub Node, or `leaf` for a Leaf Node, by using the `crsctl set node role` command. Explicitly setting the node role ensures the node type. The example in the slide shows the node `host02` being configured as a Leaf Node. Note that node role changes do not take effect until the next time that Oracle Clusterware is started on the node. This is evident in the example in the slide, where the configured node role is `leaf` but the active node role is still `hub` immediately after the node role change.
- A system administrator can also set the node role to `auto` by using the `crsctl set node role` command. This setting allows the cluster to decide which role a node will perform based on the composition of the cluster. The cluster administrator must ensure that the node can fulfill either role, `hub` or `leaf`, in order to use the `auto` setting.

Configuring the Hub Size

- Showing the current hub size:

```
$ crsctl get cluster hubsize
CRS-4950: Current hubsize parameter value is 32
```

- Setting the hub size:

```
# crsctl set cluster hubsize <number>
```

```
# crsctl set cluster hubsize 16
# crsctl get cluster hubsize
CRS-4950: Current hubsize parameter value is 16
```

ORACLE

The `auto` node role setting works in conjunction with the cluster `hubsize` setting. When a node with the `auto` node role setting attempts to join the cluster, Oracle Clusterware examines the `hubsize` setting. If the number of Hub Nodes is smaller than the `hubsize` setting, the node joins the cluster as a Hub Node. Otherwise, the node joins the cluster as a Leaf Node.

The examples in the slide show how administrators can examine and set the cluster hub size. Note that setting the cluster hub size requires system administrator privileges. Note also that the hub size setting is effective immediately but does not affect the node role of any nodes that are already in the cluster.

Configuring Miss Count for Leaf Nodes

- Viewing and setting `leafmisscount`:

```
# crsctl get css leafmisscount
CRS-4678: Successful get leafmisscount 30 for Cluster
Synchronization Services.
# crsctl set css leafmisscount 45
CRS-4684: Successful set of parameter leafmisscount to 45 for
Cluster Synchronization Services.
# crsctl get css leafmisscount
CRS-4678: Successful get leafmisscount 45 for Cluster
Synchronization Services.
```

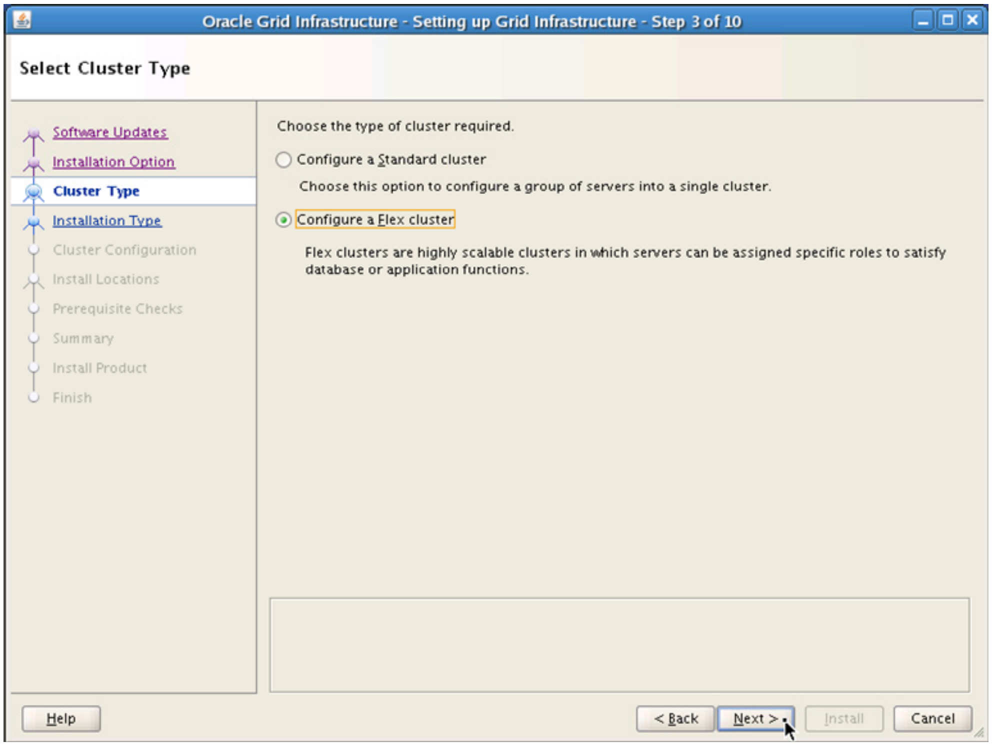

The `leafmisscount` attribute determines how Cluster Synchronization Services (CSS) handles network connectivity issues between a Leaf Node and the Hub Node that connects it to the cluster.

The `leafmisscount` setting defines the threshold duration (in seconds) for tolerable communication failures. If communication between a Hub Node and associated Leaf Node is interrupted and restored before the amount of time specified by `leafmisscount`, the cluster continues to operate normally. If communication is lost for a period exceeding the `leafmisscount` setting, the interruption is assumed to be significant and the Leaf Node is evicted from the cluster. The default `leafmisscount` setting is 30 seconds.

The examples in the slide show how to query and set the `leafmisscount` attribute.

Note that the `leafmisscount` attribute is separate from the `misscount` attribute which existed in previous releases and which continues to exist in release 12.1.

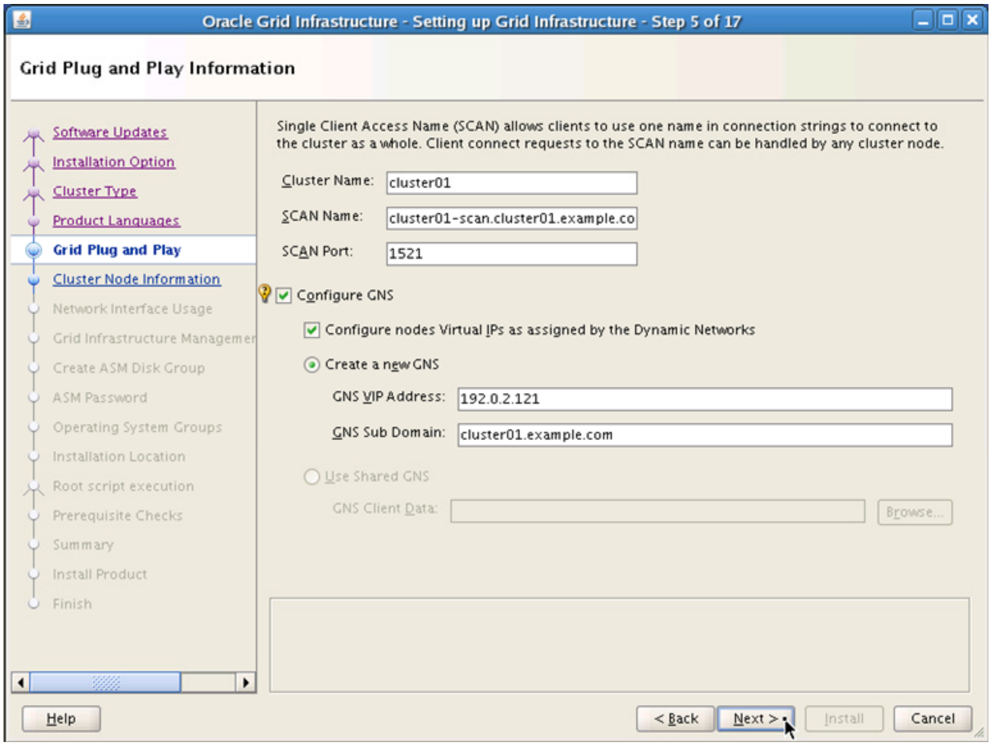
Configuring a Flex Cluster with OUI: Selecting the Cluster Type



The previous four pages introduced the commands required to configure a Flex Cluster and convert a standard cluster to a Flex Cluster. For new clusters, Oracle Universal Installer (OUI) has been updated to facilitate the configuration of Flex Clusters.

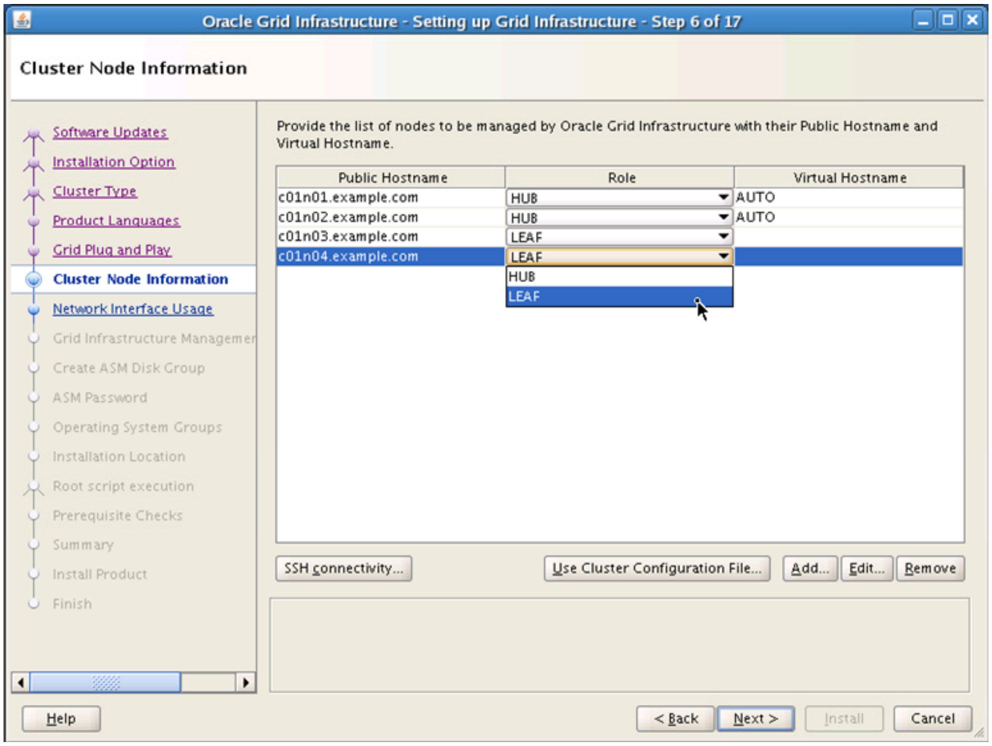
The screenshot in the slide shows the OUI interface for the step where administrators select the cluster type. To configure a Flex Cluster, administrators must select the “Configure a Flex cluster” option.

Configuring a Flex Cluster with OUI: Configuring GNS



Flex Clusters require the Grid Naming Service (GNS) to be configured with a fixed virtual IP (VIP) on one of the Hub Nodes. In line with this, OUI will not progress before GNS is configured on the Grid Plug and Play Information screen. The screenshot in the slide shows an example of the required configuration.

Configuring a Flex Cluster with OUI: Selecting the Node Type



When “Configure a Flex cluster” is selected in the Select Cluster Type screen, the Cluster Node Information screen looks like the example shown in this slide. In this interface, the user may specify the cluster nodes. For each cluster node, the user can set the node type to HUB or LEAF.

A cluster that is created with this method is configured with nodes that have explicitly defined roles.

To configure the cluster by using the `auto` node role setting in conjunction with the cluster `hubsize` setting, administrators must wait until after the cluster is initially configured with OUI. Then, they can use the commands introduced earlier in this lesson to adjust the node role and cluster `hubsize` settings.

Flex Clusters and Node Failure

- Nodes that are evicted from the cluster do not require a restart; they need only a cluster software restart.
- If a Hub Node fails:
  - The node is evicted from the cluster
    - Services are relocated to other Hub Nodes if possible.
  - Corresponding Leaf Nodes are also evicted from the cluster
    - Services are relocated to other Leaf Nodes if possible.
- If a Leaf Node fails:
  - The node is evicted from the cluster
    - Services are relocated to another Leaf Node if possible.
  - The impact of the failure is contained if possible

You have already learned about the effect of network connectivity issues between Hub Nodes and Leaf Nodes. You now learn about the effect of node failure in Flex Clusters.

In previous releases of Oracle Clusterware, any node that was evicted from the cluster would trigger panic in the operating system kernel and cause an immediate shutdown of the node. This measure is very effective at protecting the cluster from the effects of a rogue node; however, restarting the node results in the cluster working with diminished capacity for many minutes. With Oracle Clusterware 12*c*, node eviction does not require a node restart. Rather, where possible, only the cluster software is restarted, which significantly reduces the amount of time that the cluster is affected.

If a Hub Node fails, the node is evicted from the cluster in essentially the same way as any node in a standard cluster. As part of dealing with the node failure, the cluster attempts to start as many services as possible on surviving cluster nodes. Leaf Nodes that were associated with the failed Hub Node are also evicted from the cluster.

If a Leaf Node fails, the node is evicted from the cluster and the cluster attempts to relocate services running on the Leaf Node to other Leaf Nodes connected to the same Hub Node. This means that the effect of a Leaf Node failure is usually contained within the group of Leaf Nodes connected to the same Hub Node. Thus, the performance and availability of the rest of the cluster is not affected by the failure of a Leaf Node.

Quiz

Identify the use cases supported by Flex Clusters:

a. Large-scale decision support databases in which parallel queries operations can be spread across database instances running on the Leaf Nodes
b. Large-scale online transaction processing (OLTP) databases in which many thousands of user connections can be spread across database instances running on the Leaf Nodes
c. Mixed environments in which databases run on the Hub Nodes and highly available application resources run on the Leaf Nodes

Answer: c

In release 12.1, Oracle Database instances are supported on Hub Nodes only. Leaf Nodes, added in 12*c*, can be used to host application services, which can leverage the high-availability framework of Oracle Clusterware.

Quiz

Flex Clusters achieve greater scalability than standard clusters because:

a. Fewer physical network connections are required between the cluster nodes
b. Leaf Nodes do not require direct access to shared storage
c. By limiting the size of the hub, contention for key clusterware resources is controlled
d. Fewer network interactions are required between the cluster nodes to maintain the cluster
e. The cluster hub size can be set to a larger value than in previous versions

Answer: c, d

The two-layered hub-and-spoke topology of a Flex Cluster achieves greater scalability because of two fundamental reasons:

1. By limiting the size of the hub, contention for key clusterware resources, such as the OCR and voting disks, does not increase significantly because of the addition of the Leaf Nodes.
2. Fewer network interactions are required between nodes in the cluster, and consequently there is less administrative network traffic, such as heartbeats, exchanged between the nodes.

Answer A is not correct because the number of physical network connections is the same for both cluster types.

Answer B is a correct statement; however, this fact does not by itself improve cluster scalability.

Answer E is not correct because the cluster hub size setting does not by itself improve cluster scalability.

Summary

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

- Flex Cluster architecture and components
- The effect of node failure in a Flex Cluster

4

Grid Infrastructure Preinstallation Tasks

Objectives

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

Preinstallation Planning

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 [1] No [2]	Yes [3]	Yes [4]
NFS	Yes	Yes	Yes	Yes	Yes
Raw Devices	No	No	No	No	No
Local File System	No	Yes	Yes	No	No

1. Yes on Standard configuration or admin-managed Oracle Database 11*g* Release 2 and later
2. No on Oracle Flex Cluster policy-managed databases because neither Oracle Database nor ACFS can run on Leaf Nodes
3. Yes for Oracle Database 12*c* Release 1 and later
4. Yes for Oracle Database 12*c* Release 1 and later

The Oracle Clusterware configuration files (voting and Oracle Cluster Registry [OCR]) can be stored on a supported cluster file system, ASM, or NFS on a certified NAS filer. Please note that Direct NFS Client does not support Oracle Clusterware files. In conjunction with ASM Cluster File System, ASM provides a complete shared storage solution for Oracle Grid Infrastructure and RAC. As noted in the table in the slide, raw or block devices are not directly supported.

If you intend to upgrade an existing Oracle RAC database, or an Oracle RAC database with ASM instances, you can use an existing raw or block device partition and perform a rolling upgrade of your existing installation.

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 choose to place the OCR files on a shared file system, ensure that:
  - The disks used for the file system are on a highly available storage device (for example, a RAID device)
  - At least two file systems are mounted, and use Oracle Clusterware 12*c* features to provide redundancy for the OCR

ORACLE

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

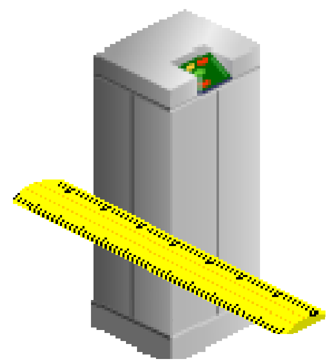
- To use an NFS file system, it must be on a supported NAS device. Log in to My Oracle Support and click Certifications to find the most current information about supported NAS device.
- If you choose to place your Oracle Cluster Registry (OCR) files on a shared file system, Oracle Corporation recommends that you configure your shared file systems in one of the following ways:
  - The disks used for the file system are on a highly available storage device, (for example, a RAID device).
  - At least two file systems are mounted, and use the features of Oracle Clusterware 12*c* Release 1 (12.1) to provide redundancy for the OCR.
- If you choose to place your database files on a shared file system, then one of the following should be true:
  - The disks used for the file system are on a highly available storage device, (for example, a RAID device).
  - The file systems consist of at least two independent file systems, with the database files on one file system, and the recovery files on a different file system.

- Oracle recommends that you create a Grid Infrastructure Management Repository. This repository is an optional component, but if you do not select this feature during installation, you lose access to Oracle Database Quality of Service management, Memory Guard, and Cluster Health Monitor. You cannot enable these features after installation except by reinstalling Oracle Grid Infrastructure.
- The user account with which you perform the installation (`oracle` or `grid`) must have write permissions to create the files in the path that you specify.

Sizing Storage for Oracle Clusterware

Assuming external redundancy:

- At least 300 MB is needed for each voting disk volume
- At least 400 MB is needed for each OCR volume
- At least 3.6 GB is needed for the OCR volume containing the Management Repository
- Add an additional 500 MB for each node for clusters larger than four nodes



To store all Oracle Clusterware files on the shared file system with external redundancy, you will need a minimum of 300 MB for each voting disk volume. In addition, You will also need a minimum of 400 MB for each OCR volume, assuming that a Grid Infrastructure Management Repository was not implemented. If a Management Repository is needed, the OCR volume storing the repository should be a minimum of 3.6 GB plus 500 MB for each node for clusters greater than four nodes. For example, a six-node cluster allocation should be 4.6 GB.

Sizing Storage for Oracle Clusterware

Assuming normal redundancy:

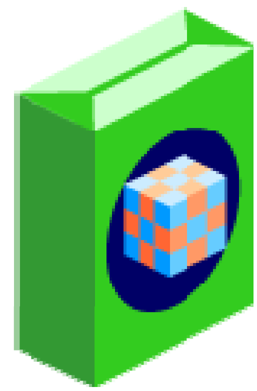
- To store all Oracle Clusterware files, you need:
  - A minimum of three volumes
  - A minimum of three physical disks and three failure groups
  - At least 2 GB of available storage
- To store Clusterware files and a Management Repository, you need a minimum of 3.7 GB.
  - 2 GB for the Management Repository
  - At least 400 MB for each OVR volume (2), totaling 800 MB
  - At least 300 MB for each voting disk (3), totaling 900 MB
- Add an additional 500 MB for each node for clusters larger than four nodes.

To store all Oracle Clusterware files on the shared file system with normal redundancy, you should have at least 2 GB of storage available over a minimum of three volumes (three separate volume locations for the OCR and two OCR mirrors, and one voting disk on each volume). You should have a minimum of three physical disks, each at least 500 MB, to ensure that voting disks and OCR files are on separate physical disks. If you will be employing a Grid Infrastructure Repository, add another 2 GB of storage plus 500 MB for each node for clusters greater than four nodes.

If you create partitions on shared partitions with `fdisk` by specifying a device size, such as `+400M`, the actual device created may be smaller than the size requested, based on the cylinder geometry of the disk. This is due to current `fdisk` restrictions. Oracle recommends that you partition the entire disk that you allocate for use by Oracle ASM.

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 Grid Infrastructure and Oracle RAC only support cluster-aware volume managers. Some third-party volume managers are not cluster-aware, and so are not supported. To confirm that a volume manager you want to use is supported, click Certifications on My Oracle Support to determine if your volume manager is certified for Oracle RAC. My Oracle Support is available at the following URL: https://support.oracle.com

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 can 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 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. If 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 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
```

Quiz

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

a. True
b. False

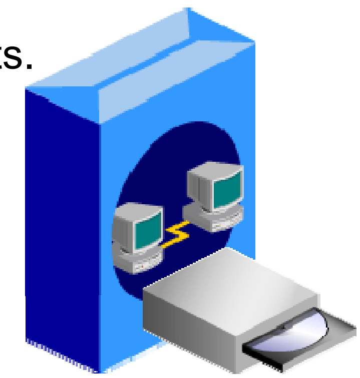
Answer: a

This statement is true.

Grid Infrastructure Preinstallation Tasks

Oracle Grid Infrastructure 12*c* 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.



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.

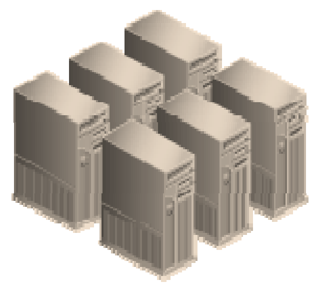
Oracle Grid Infrastructure must be installed by using the graphical OUI. Character-based tool installations are not possible; however, OUI can be run in silent mode using response files to supply the values that the installation would need. Ensure that your cluster hardware is functioning normally before 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.



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 display cards provide at least 1024 x 768 display resolution, so that OUI displays correctly while performing a system console–based installation.
- Ensure 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 using different configuration, that you categorize cluster nodes into homogenous pools as part of your server categorization management policy.

Checking System Requirements

- At least 4 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 8 GB of local storage is needed for the software.

```
]# 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.

The system must meet the following minimum hardware requirements:

- The minimum required RAM is 4 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). To determine the amount of disk space available in the `/tmp` directory, enter the following command:

    `df -h /tmp`.
- At least 8 GB is required on each node for the Grid Infrastructure software home.

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 12*c*, 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 `nscd` is on for both run levels, execute the following command:

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


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

To check to see if `nscd` 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, `nscd` is enabled for run level 3, and disabled for run level 5. The `nscd` should be enabled for both run level 3 and run level 5.

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

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

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
```


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/grug/grub.conf` file

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

Single-Client Access Name for the Cluster

- If you configure a Standard cluster and choose a Typical install, then the SCAN is also the name of the cluster.
  - The SCAN must meet the requirements for a cluster name, so the SCAN can be no longer than 15 characters.
- In an Advanced installation, the SCAN and cluster name are entered in separate fields, so the SCAN can be greater than 15 characters.
- If you enter a domain with the SCAN name and want to use GNS with zone delegation, the domain must be the GNS domain.
- The SCAN must be globally unique in your enterprise.
- After installation, you can change the cluster name only by reinstalling Oracle Grid Infrastructure.

ORACLE

The cluster name 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.

If you configure a Standard cluster, and choose a Typical install, the SCAN is also the name of the cluster. In that case, the SCAN must meet the requirements for a cluster name. The SCAN can be no longer than 15 characters. In an Advanced installation, 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.

Select your 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
```


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.

If you enable Grid Naming Service (GNS), then name resolution requests to the cluster are delegated to the GNS, which is listening on the GNS virtual IP address. You define this address in the DNS domain before installation. The DNS must be configured to delegate resolution requests for cluster names (any names in the subdomain delegated to the cluster) to the GNS. When a request comes to the domain, GNS processes the requests and responds with the appropriate addresses for the name requested.

To use GNS, before installation, the DNS administrator must establish DNS Lookup to direct the DNS resolution of a subdomain to the cluster. If you enable GNS, then you must have a DHCP service on the public network that allows the cluster to dynamically allocate the virtual IP addresses as required by the cluster.

Note: If you have vendor clusterware installed, then you cannot choose to use GNS, because vendor clusterware does not support it.

IP Address Requirements for Static Configuration

- If you do not enable GNS, 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

If you do not enable GNS, 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

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.

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.

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 must use IPv4 addresses 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.

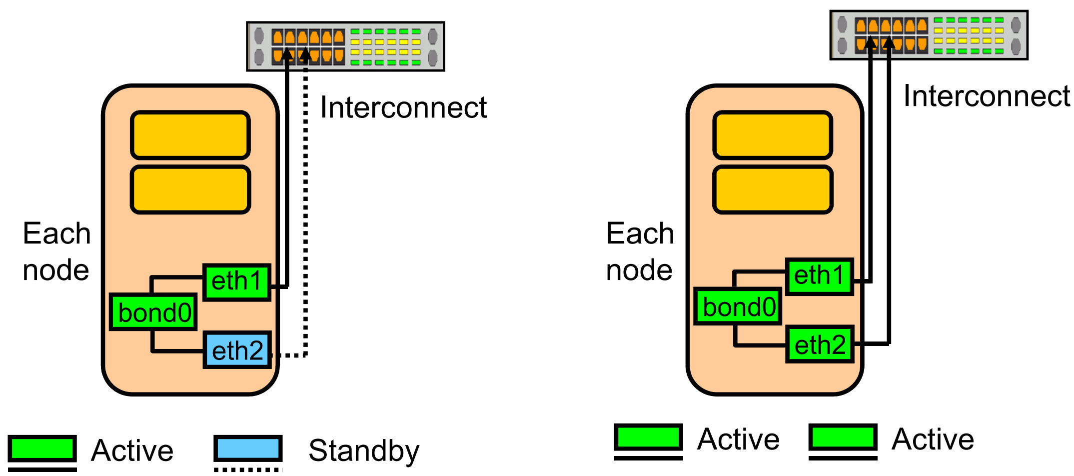
With Redundant Interconnect Usage, you can identify multiple interfaces to use for the cluster private network, without the need of using bonding or other technologies. This functionality is available starting with Oracle Database 11*g* Release 2 (11.2.0.2). If you use the Oracle Clusterware Redundant Interconnect feature, you must use IPv4 addresses for the 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.



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.

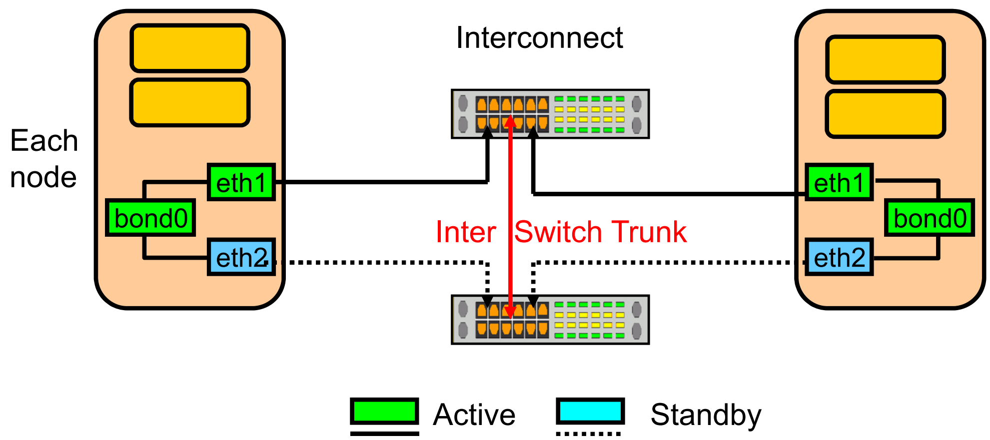
Note: During the installation of the Oracle Clusterware product, the `bond0` interface will be supplied as the value to the prompts for the interconnect interface to be used.

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.



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)/*i*SCSI storage
- Have network device interoperability concerns
- Need to be configured with care and tested rigorously

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.

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 deinstall 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/ntpd.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 ctss
```

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 128ms, even under extreme cases of network path congestion and jitter. Sometimes, in particular when `ntpd` is first started, the error might exceed 128ms. This may on occasion cause the clock to be stepped backward if the local clock time is more than 128ms 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 6: 2.6.32-100.28.5.el6.x86_64 or later Oracle Linux 6 with the Red Hat Compatible kernel: 2.6.32-71.el6.x86_64 or later Oracle Linux 5 Update 6 with the Unbreakable Enterprise kernel: 2.6.32-100.0.19 or later Oracle Linux 5 Update 6 with the Red Hat compatible kernel: 2.6.18-238.0.0.0.1.el5
Red Hat Enterprise Linux (RHEL)	RHEL 6: 2.6.32-71.el6.x86_64 or later RHEL 6 with the Unbreakable Enterprise Kernel: 2.6.32-100.28.5.el6.x86_64 or later RHEL 5 Update 6 with the Unbreakable Enterprise Kernel: 2.6.32-100.0.19 or later or later Red Hat Enterprise Linux 5 Update 6: 2.6.18-238.el5 or later
SUSE Enterprise Linux	SUSE Linux Enterprise Server 11 2.6.27.19 or later

For installations of Oracle Grid Infrastructure, ensure that you have the distributions and kernel versions listed in the table in the slide. The Linux distributions and packages listed in this lesson are supported for this release on x86-64. No other Linux distributions are supported.

If you intend to install Oracle Database or Oracle RAC in addition to Oracle Grid Infrastructure, refer to the *Oracle Grid Infrastructure Installation Guide, 12c Release 1 (12.1) for Linux* to determine whether you must install additional packages for the features you plan to use.

To obtain the most current information about kernel requirements, see the online version on the Oracle Technology Network at the following URL:

http://www.oracle.com/technetwork/indexes/documentation/index.html

Software Requirements: Packages

Linux x86 Oracle 64-bit Grid Infrastructure and Oracle RAC RPM requirements for OL 5 and RHEL 5

```
binutils-2.17.50.0.6
compat-libstdc++-33-3.2.3
compat-libstdc++-33-3.2.3 (32 bit)
gcc-4.1.2
gcc-c++-4.1.2
glibc-2.5-58
glibc-2.5-58 (32 bit)
glibc-devel-2.5-58
glibc-devel-2.5-58 (32 bit)
ksh
libaio-0.3.106
libaio-0.3.106 (32 bit)
libaio-devel-0.3.106
libaio-devel-0.3.106 (32 bit)
libgcc
libgcc-4.1.2 (32 bit)
libstdc++-4.1.2
libstdc++-4.1.2 (32 bit)
libstdc++-devel 4.1.2
libXext-1.0.1
libXext-1.0.1 (32 bit)
libXtst-1.0.1
libXtst-1.0.1 (32 bit)
libX11-1.0.3
libX11-1.0.3 (32 bit)
libXau-1.0.1
libXau-1.0.1 (32 bit)
libXi-1.0.1
libXi-1.0.1 (32 bit)
make-3.81
sysstat-7.0.2
```


For Linux86 64-bit systems running OL 5, or RHEL 5, 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 6 and RHEL 6

```
binutils-2.20.51.0.2-5.11.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
gcc-4.4.4-13.el6 (x86_64)
gcc-c++-4.4.4-13.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 (x86_64)
libstdc++-4.4.4-13.el6 (x86_64)
libstdc++-devel-4.4.4-13.el6 (x86_64)
libstdc++-devel-4.4.4-13.el6.i686
libaio-devel-0.3.107-10.el6 (x86_64)
libaio-0.3.107-10.el6 (x86_64)
libaio-devel-0.3.107-10.el6.i686
libstdc++-4.4.4-13.el6.i686
libgcc-4.4.4-13.el6 (i686)
libaio-0.3.107-10.el6.i686
libXext-1.1 (x86_64)
libXext-1.1 (i686)
libXtst-1.0.99.2 (x86_64)
libXtst-1.0.99.2 (i686)
libX11-1.3 (x86_64)
libX11-1.3 (i686)
libXau-1.0.5 (x86_64)
libXau-1.0.5 (i686)
libxcb-1.5 (x86_64)
libxcb-1.5 (i686)
libXi-1.3 (x86_64)
libXi-1.3 (i686)
make-3.81-19.el6
sysstat-9.0.4-11.el6 (x86_64)
```


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 SUSE Linux Enterprise Server 11

```
binutils-2.21.1-0.7.25
gcc-4.3-62.198
gcc-c++-4.3-62.198
glibc-2.11.3-17.31.1
glibc-devel-2.11.3-17.31.1
ksh-93u-0.6.1
libaio-0.3.109-0.1.46
libaio-devel-0.3.109-0.1.46
libcap1-1.10-6.10
libstdc++33-3.3.3-11.9
libstdc++33-32bit-3.3.3-11.9
libstdc++43-devel-4.3.4_20091019-0.22.17
libstdc++46-4.6.1_20110701-0.13.9
libgcc46-4.6.1_20110701-0.13.9
make-3.81
sysstat-8.1.5-7.32.1
xorg-x11-libs-32bit-7.4
xorg-x11-libs-7.4
xorg-x11-libX11-32bit-7.4
xorg-x11-libX11-7.4
xorg-x11-libXau-32bit-7.4
xorg-x11-libXau-7.4
xorg-x11-libxcb-32bit-7.4
xorg-x11-libxcb-7.4
xorg-x11-libXext-32bit-7.4
xorg-x11-libXext-7.4
```


For Linux x86 64-bit systems running SUSE Linux Enterprise Server 11, the packages listed in the slide (or later versions) must be installed.

Oracle Linux and 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.

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.

Oracle Pre-Install RPM

- Use the Oracle RDBMS Server 12*c*R1 Pre-Install RPM to complete most configuration tasks for OL and RHEL.
- When it is installed, the Oracle Pre-Install 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
- For Oracle Linux Release 5.2 and higher, the Oracle Pre-Install RPM is included on the install media.

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 RDBMS Server 12cR1 Pre-Install RPM, available from the Oracle Linux Network, or available on the Oracle Linux DVDs. Using the Oracle Pre-Install 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 Pre-Install 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 Pre-Install 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 Pre-Install RPM does not install OpenSSH, which is required for Oracle Grid Infrastructure installation. If you perform a minimal Linux installation and install the Oracle Pre-install 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 Pre-Install 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-1.0.9-1.rpm
```


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, deinstall 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-1.0.9-1.rpm
```

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 Kspice and Upstart RPMs with the commands below:

```
# rpm -i ksplice-uptrack-release.noarch.rpm
# yum -y install uptrack
```

  4. Execute the uptrack-upgrade command as the `root` user:

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


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 downtime update of your kernel:
```
# uptrack-upgrade -y
```

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

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
```


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 54324 asmdba
# /usr/sbin/groupadd -g 54327 asmadmin
# /usr/sbin/groupadd -g 54328 asmoper
# /usr/sbin/groupadd -g 54329 asmadmin
```

2 Create Users

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


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, oracle
# /usr/sbin/useradd -u 1101 -g oinstall -G asmdba,dba,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
```


The recommendation for the Oracle Grid Infrastructure owner is to increase the hard limits for the maximum number of processes (`nproc`) and the maximum number of open files (`nofile`) to higher values than the default values. The hard limits are set by the superuser and enforced by the kernel. End-user accounts are not allowed to raise their limits above the hard settings. The soft limits are considered to be the default starting values for normal usage, but end users may alter the soft limits.

Note: These requirements also exist for the account, typically named `oracle`, that will be used for the database software. Consider adding this to `/etc/security/limits.conf` in addition to the Oracle Grid Infrastructure account owner.

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 12*c*, 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.

During Oracle Grid Infrastructure installation, the installer requires you to run scripts with `root` privileges to complete a number of system configuration tasks. You can continue to run scripts manually as `root`, or you can delegate to the installer the privilege to run configuration steps as root, using one of the following options:

- Use the `root` password: Provide the password to the installer as you are providing other configuration information. The password is used during installation, and not stored. The root user password must be identical on each cluster member node. To enable `root` command delegation, provide the `root` password to the installer when prompted.
- Use `sudo`: Sudo is a UNIX and Linux utility that allows members of the sudoers list privileges to run individual commands as `root`. To enable sudo, have a system administrator with the appropriate privileges configure a user that is a member of the sudoers list, and provide the username and password when prompted during installation.

Quiz

Which of the following statements is *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 software is needed on each node.

Answer: b

Statement b is not true. The amount of swap space needed is determined by the amount of system memory. Systems with 4-16 GB of memory should have 1 times system memory available for swap. Systems with more than 16 GB of memory should have 16 GB available for swap.

Summary

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

5

Grid Infrastructure Installation

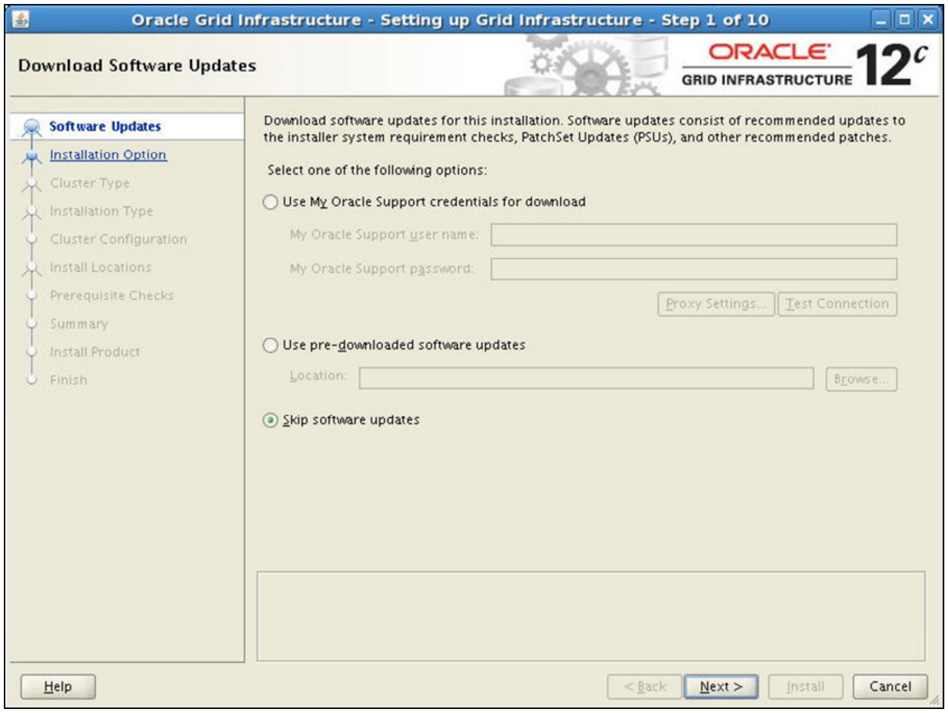
Objectives

After completing this lesson, you should be able to:

- Install Grid Infrastructure
- Verify the installation
- Configure Automatic Storage Management (ASM) disk groups

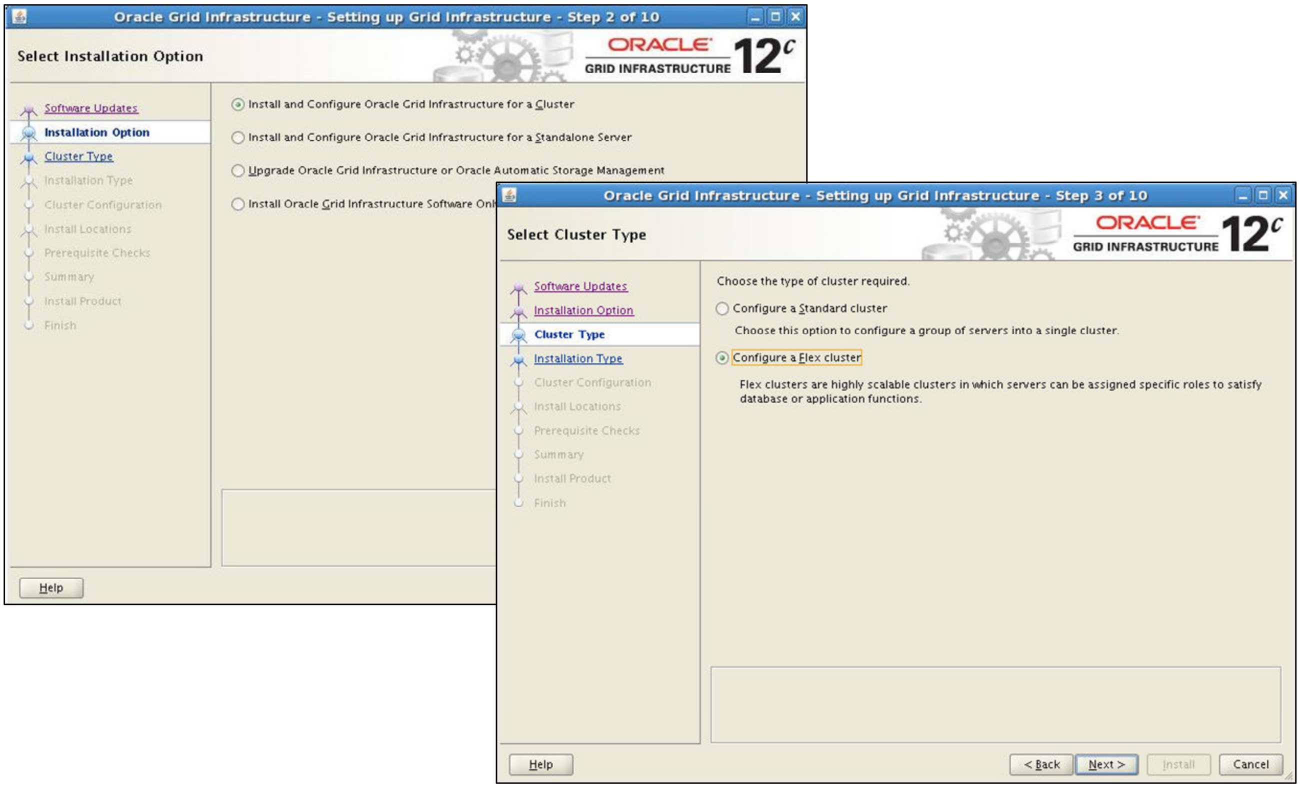
Installing Grid Infrastructure

```
./runInstaller
```



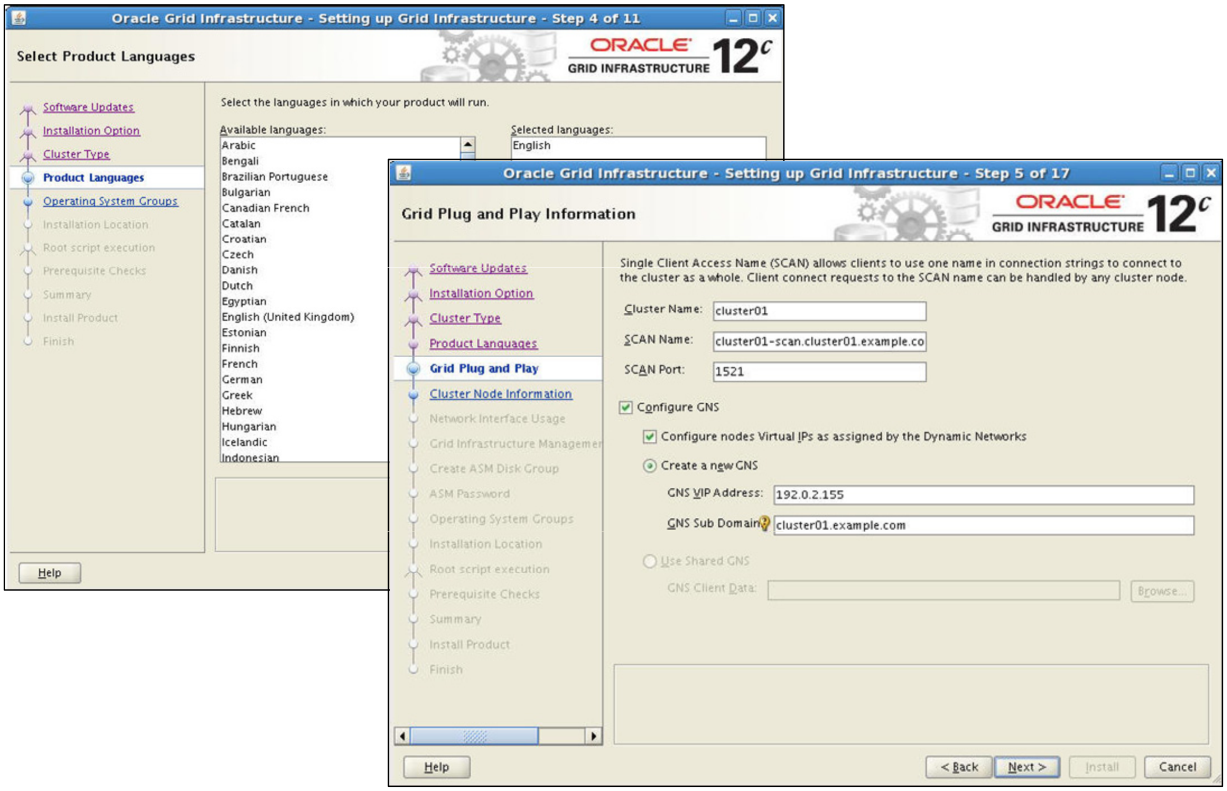
Run OUI by executing the `runInstaller` command as the grid owner from the installation CD or the staged software directory. If you have any updates or patches to include with the installation, you can specify the appropriate action on the Download Software Updates page. From here, you can provide your Oracle Support login credentials and download patches for this installation. You can also specify a location where updates have been previously staged. In the example in the slide, the user has chosen to skip software updates.

Choosing an Installation and Cluster Type



On the Select Installation Option page, select the “Install and Configure Grid Infrastructure for a Cluster” option and click Next. On the Select Cluster Type page, you can choose to install and configure either a Standard cluster or a Flex cluster. Click the cluster type required for the installation and click Next.

Grid Plug and Play Support



ORACLE

After you have chosen product languages from the Select Product Languages page, 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 connecting to the cluster. The SCAN is a domain name registered to three IP addresses, either in the domain name server (DNS) or the Grid Naming Service (GNS). The 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.

If you want to employ GNS, select the Configure GNS check box. You can choose to have DHCP assign VIP addresses by selecting the “Configure nodes Virtual IPs as assigned by the Dynamic Networks” check box. There are two GNS configuration options from which to choose. If you would like to create a new GNS for the cluster, select “Create a new GNS”. Provide the GNS VIP address and GNS subdomain. Specifying the Cluster Name and the GNS subdomain, the SCAN defaults to *clustername-scan.GNSdomain*. For example, if you start the Grid Infrastructure installation from `host01` and the cluster name is `cluster01` and the GNS domain is `cluster01.example.com`, then the SCAN becomes `cluster01-scan.cluster01.example.com`. Select “Use Shared GNS” to use an existing GNS.

Configuring Shared GNS

1. Export GNS credentials on the GNS server cluster.

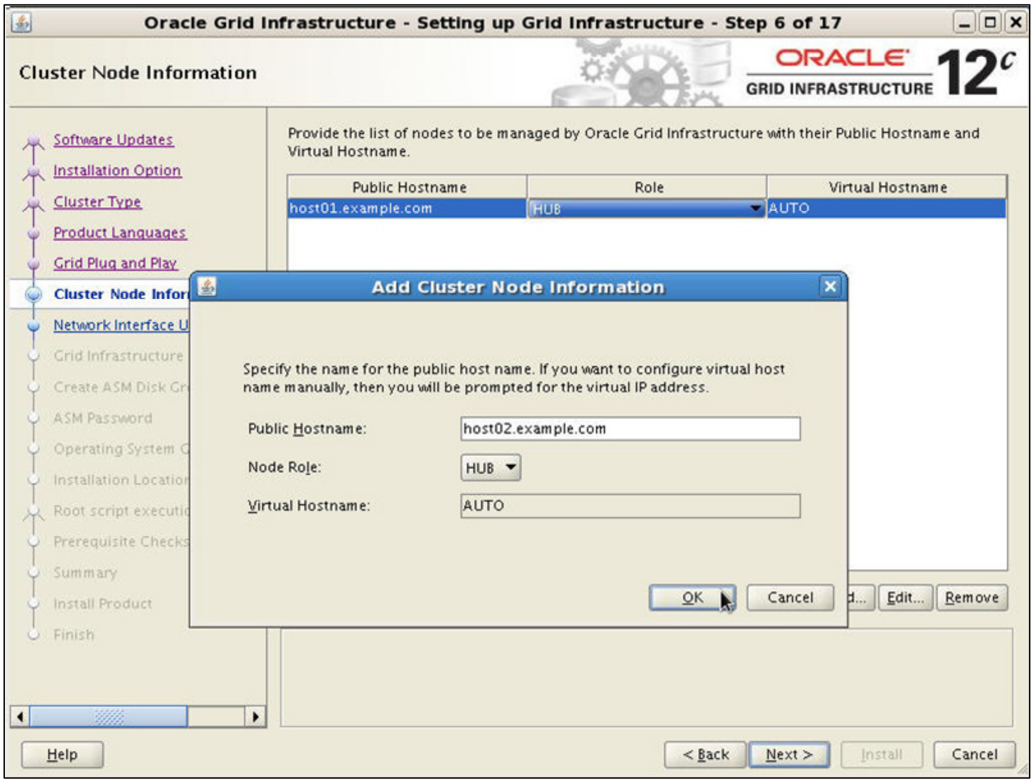
```
# srvctl export gns -clientdata /tmp/gnsclientdata
```

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 GNS client data exported from another host 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 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



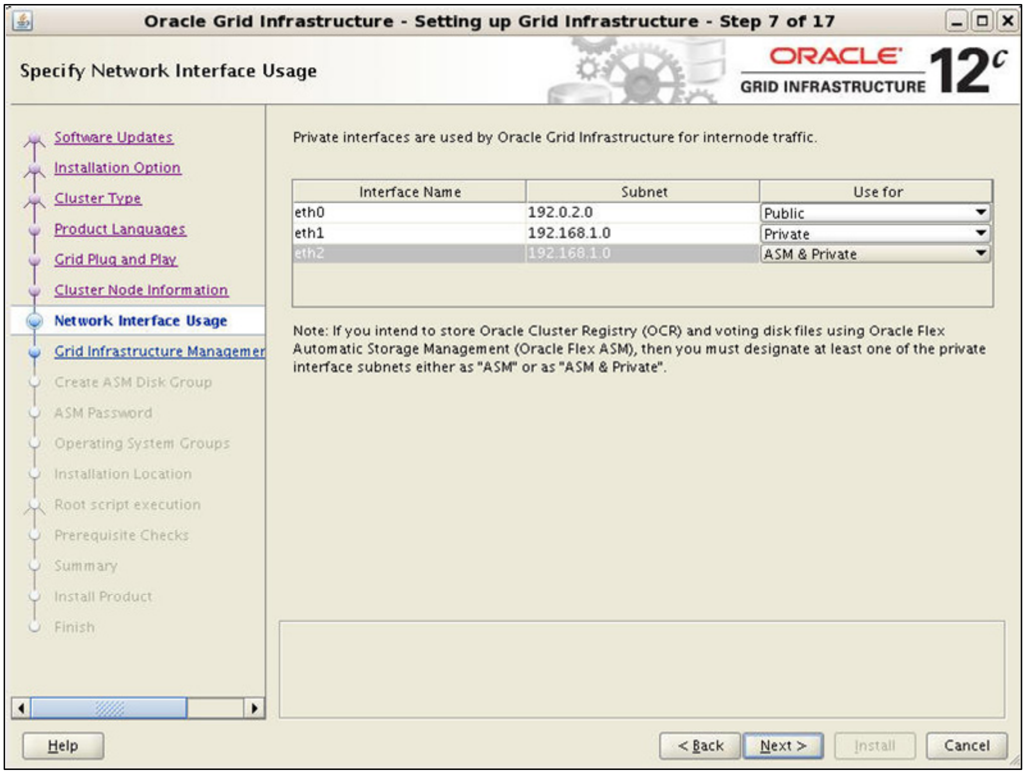
Cluster Node Information: Advanced Installation

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 has been 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 using IPs assigned by DHCP. If you are creating a Flex Cluster, you will need to indicate whether the node you are adding will be a Hub or Leaf node. Use the pull-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 common to all nodes. When SSH connectivity has been established, click Next to continue.

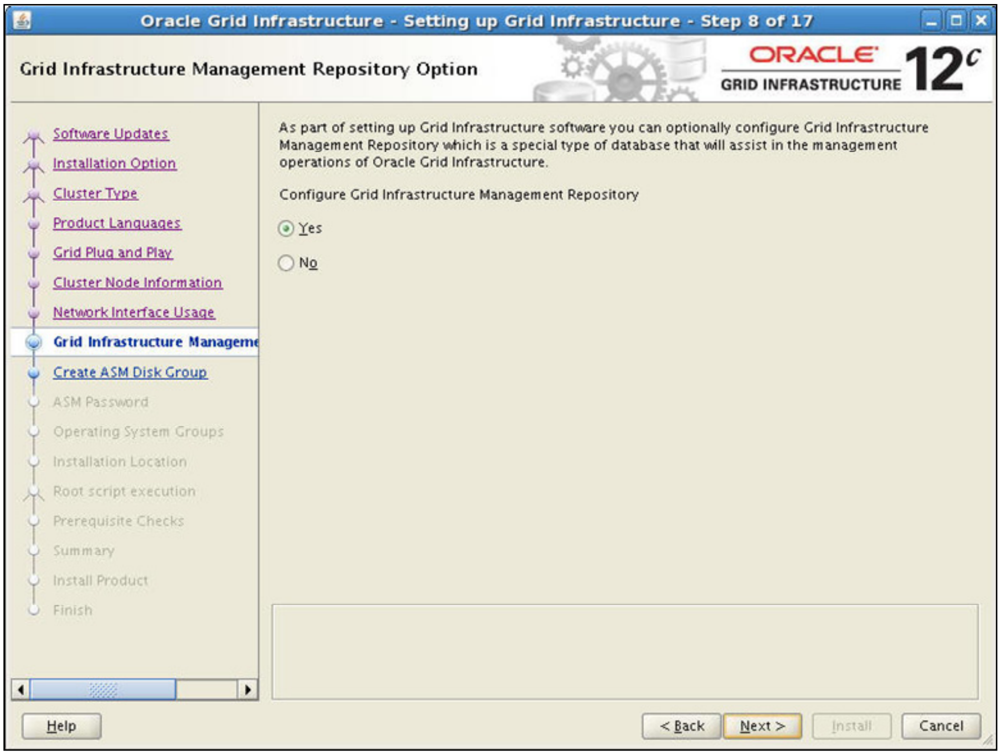
Specify Network Interface Usage



Specify Network Interface Usage: Advanced Installation

On the Specify Network Interface Usage page, you can select the network interfaces on your cluster nodes to use for internode communication. If you are creating a Standard cluster, you must choose one interface for the public network and one for the private network. If you are creating a Flex cluster, you must choose one interface for the public network and one for 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 pull-down list to the right of the interface, under the "Use for" column and choose the proper usage for each network interface. In the example shown in the slide, there are three interfaces: eth0, eth1, and eth2. The eth0 interface is the hosts' primary network interface and should be marked Public. The eth1 interface is configured for the private interconnect and is marked Private. The third adapter is used for the private network and ASM. 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.

Grid Management Repository Option

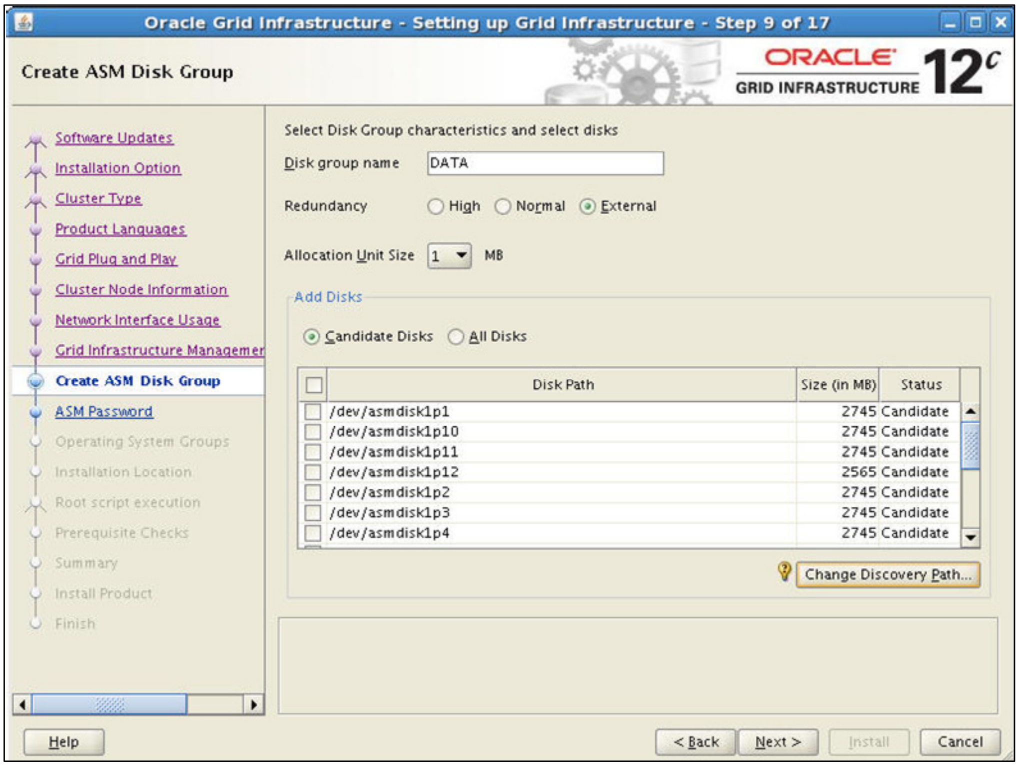


On the Grid Infrastructure Management Repository Option page, you can choose to create a Grid Management Repository database. This database is an enhancement of Cluster Health Monitor. The CHM repository, which previously resided in a file-based database, is implemented inside a dedicated Oracle database known as the Grid Infrastructure Management Repository. This enables the CHM repository to scale in size beyond previous limits and support longer retention periods for CHM data. Also, moving the CHM repository into an Oracle database enables integration of CHM data with Oracle Enterprise Manager, which in turn allows CHM data to be correlated with other metrics inside Enterprise Manager.

The CHM architecture is enhanced to support the Hub Node and Leaf Node architecture of Flex Clusters. This enables CHM to perform effectively with clusters containing hundreds, and perhaps thousands, of nodes.

Oracle recommends that you create a Grid Infrastructure Management Repository. This repository is an optional component, but if you do not select this feature during installation, you lose access to Oracle Database Quality of Service management, Memory Guard, and Cluster Health Monitor. You cannot enable these features after installation except by reinstalling Oracle Grid Infrastructure. To create a Grid Infrastructure Management Repository, select the Yes option and click Next.

Create ASM Disk Group



On the Create ASM Disk Group page, you can configure the default DATA disk group used to store the OCR and voting disks. Here, you can change the disk group name, set the redundancy and 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 you put in the disk group. Normal redundancy disk groups provide three voting disk files, one OCR and two copies (one primary and one secondary mirror). With normal redundancy, the cluster can survive the loss of one failure group. For most installations, Oracle recommends that you select normal redundancy. A normal redundancy disk group requires a minimum of two disk devices (or two failure groups). If storing OCR and voting disks, that number would be three. The effective disk space in a normal redundancy disk group is half the sum of the disk space in all of its devices. A normal redundancy disk group requires a minimum of two disk devices (or two failure groups). The effective disk space in a normal redundancy disk group is half the sum of the disk space in all of its devices.

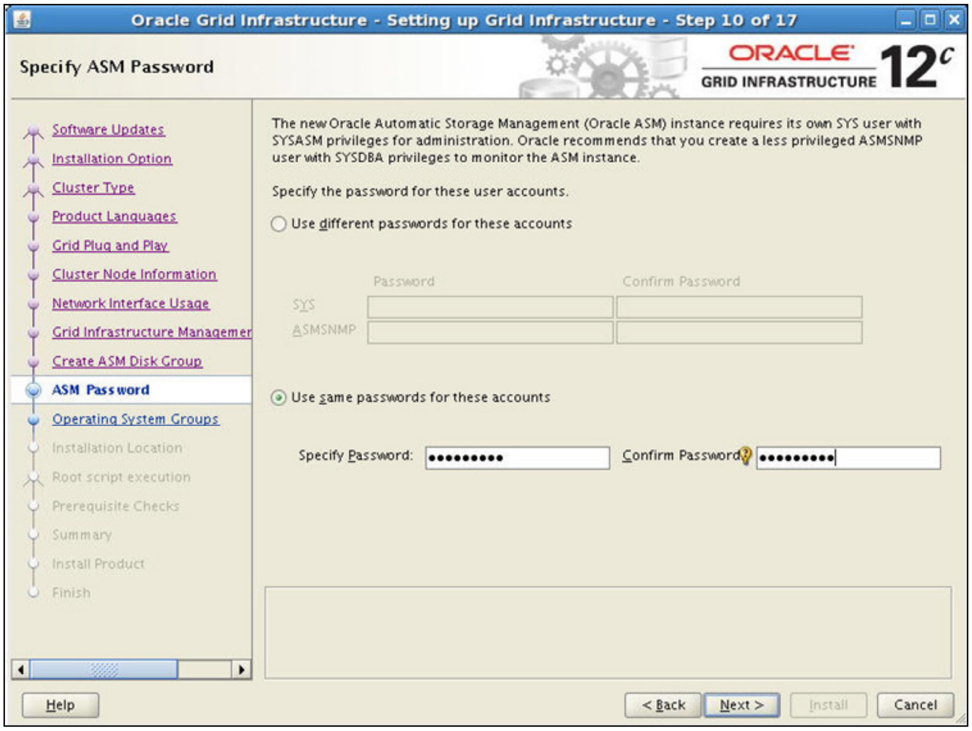
High redundancy disk groups provide five voting disk files, one OCR and three copies (one primary and two secondary mirrors). With high redundancy, the cluster can survive the loss of two failure groups. Although high redundancy disk groups do provide a high level of data protection, you should consider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

In a high redundancy disk group, Oracle ASM uses three-way mirroring to increase performance and provide the highest level of reliability.

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.

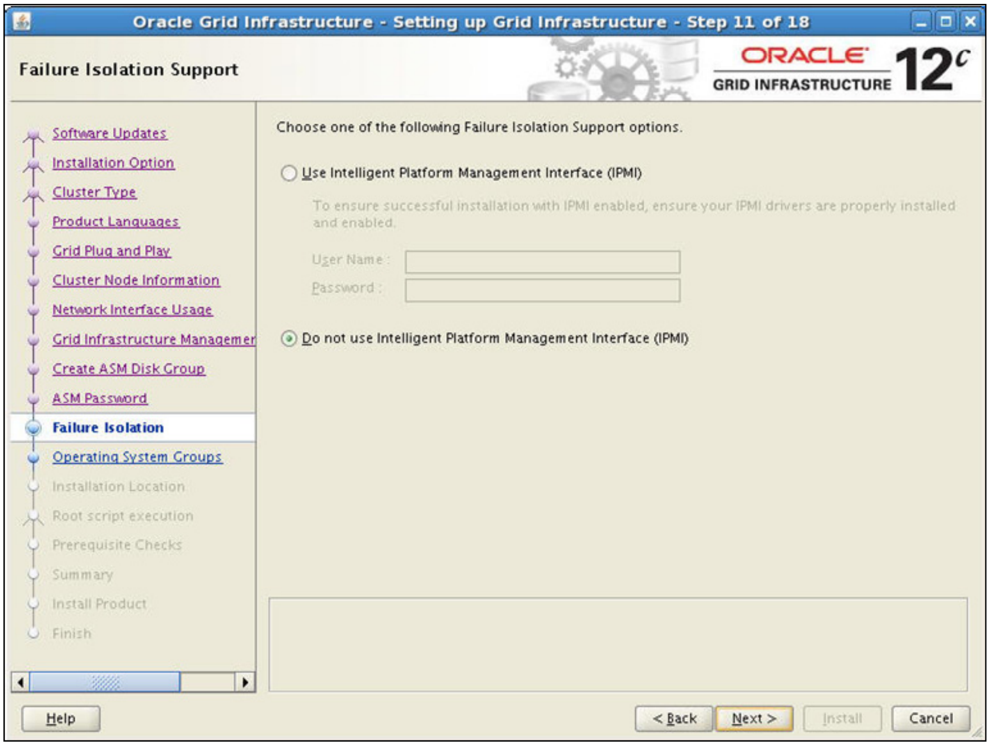
When you have made all of the necessary selections, click Next to continue.

Specify ASM Password



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



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.

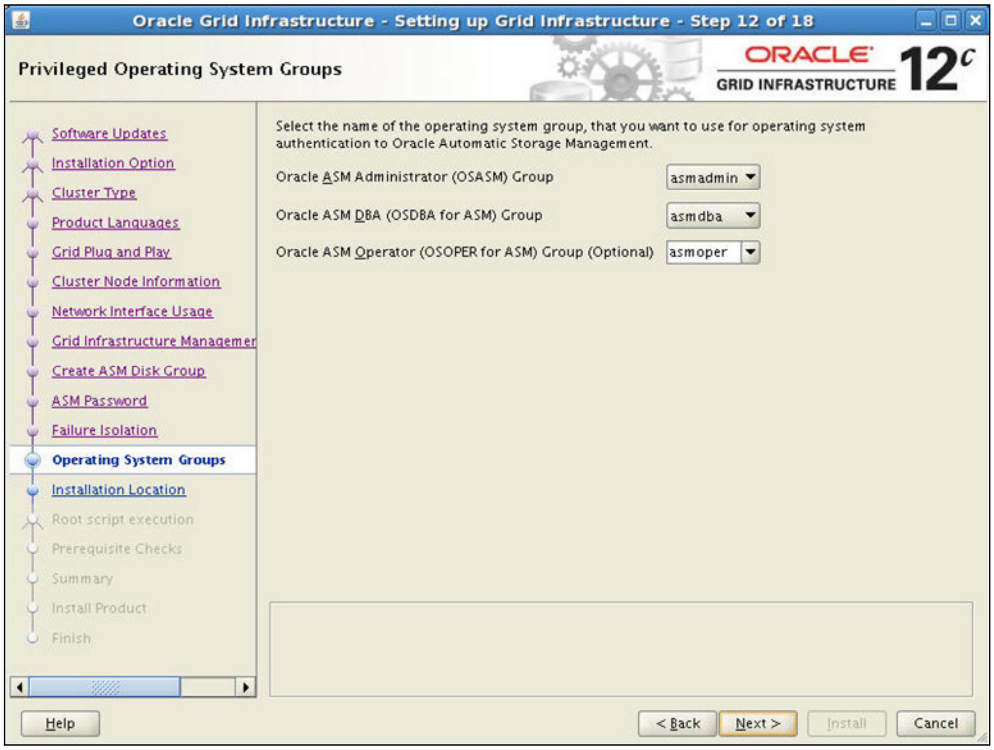
You must have the following hardware and software configured to enable cluster nodes to be managed with IPMI:

- Each cluster member node requires a Baseboard Management Controller (BMC) running firmware compatible with IPMI version 1.5 or higher, 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 to provide when prompted during installation.

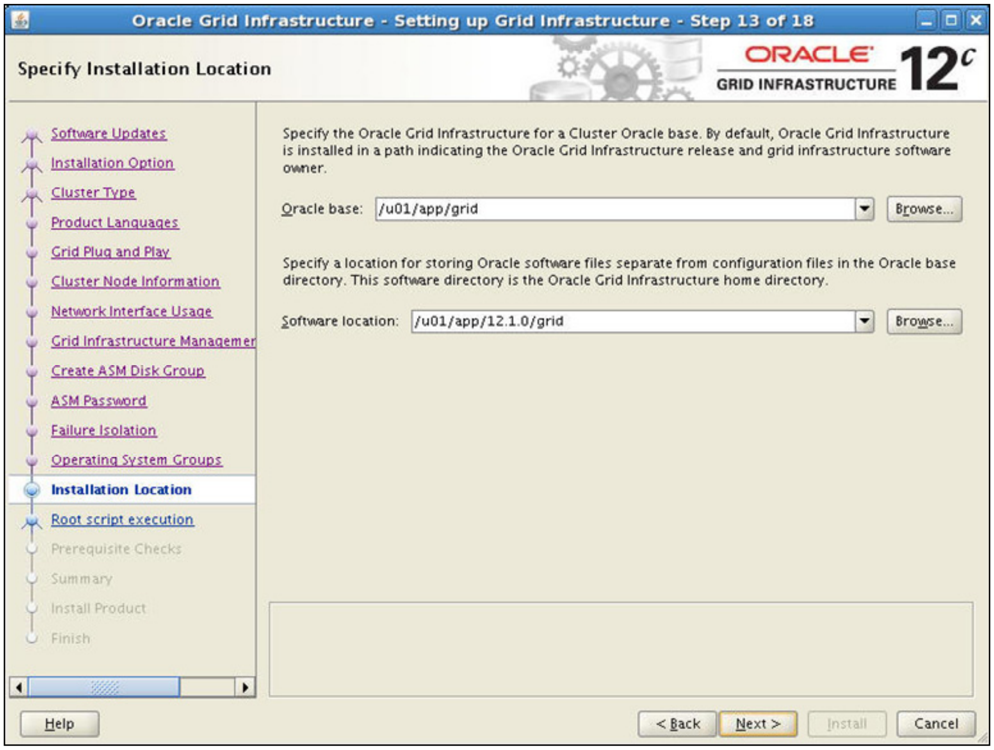
Note: For Oracle Clusterware to communicate with BMC, the IPMI driver must be installed permanently on each node, so that it is available on system restarts.

Privileged Operating System Groups



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.

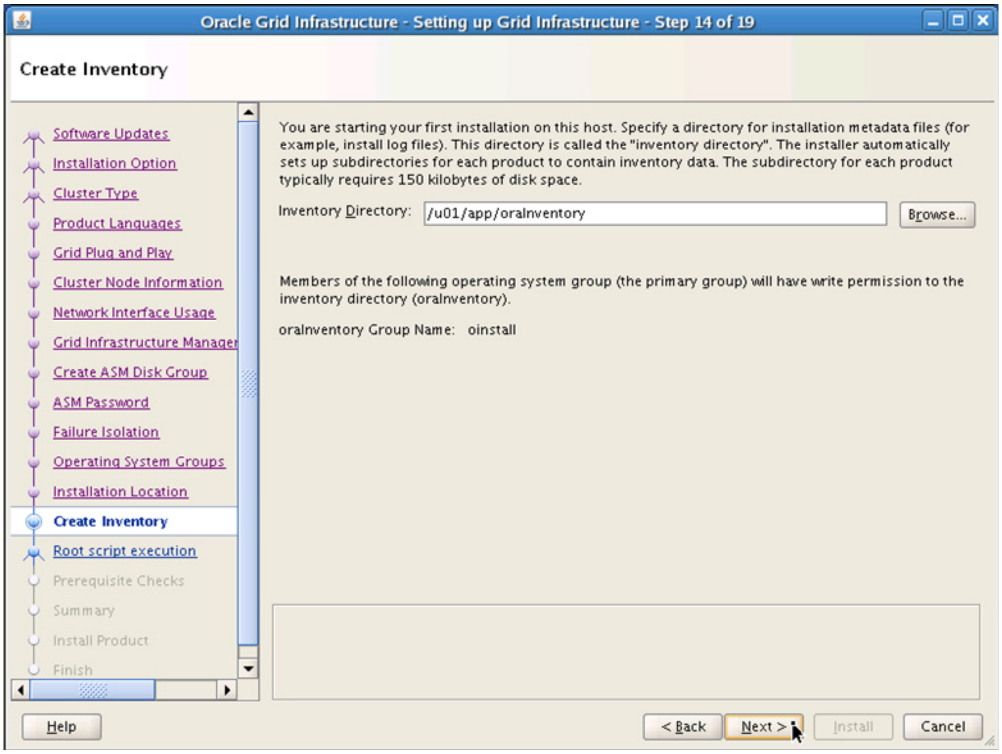
Specify Installation Location



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

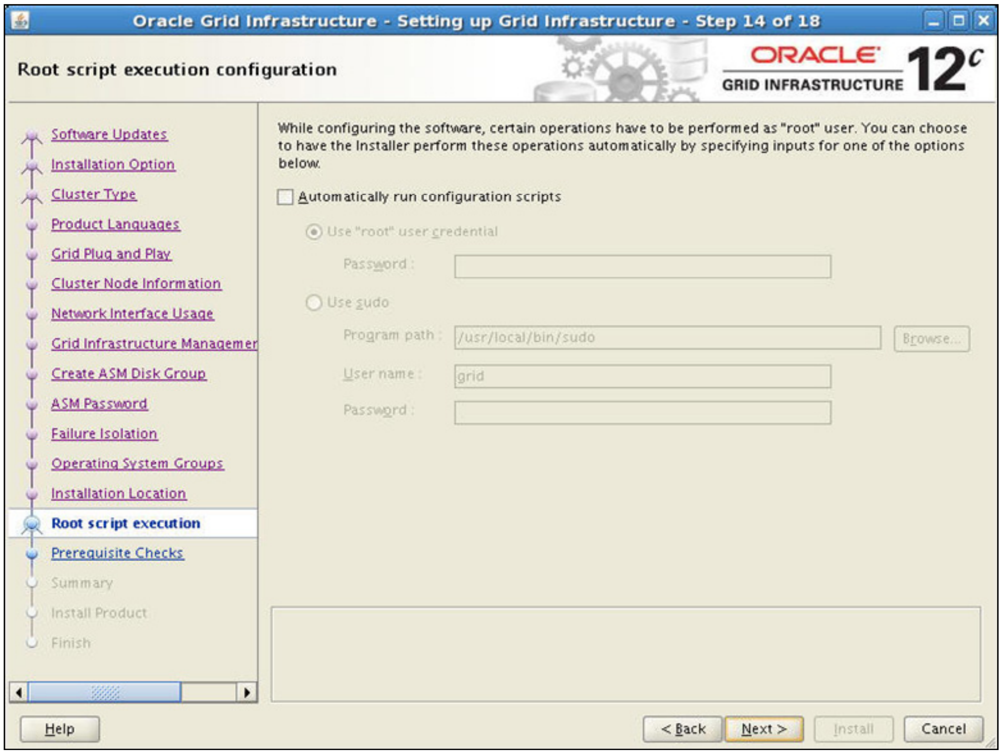
In the Software location field, enter a directory in which to install Oracle Grid Infrastructure. In the example, the grid operating system user will own this installation, so the software location should be `/u01/app/12.1.0/grid`. When you have entered proper Oracle Base and Software Location values, click Next to continue.

Create Inventory



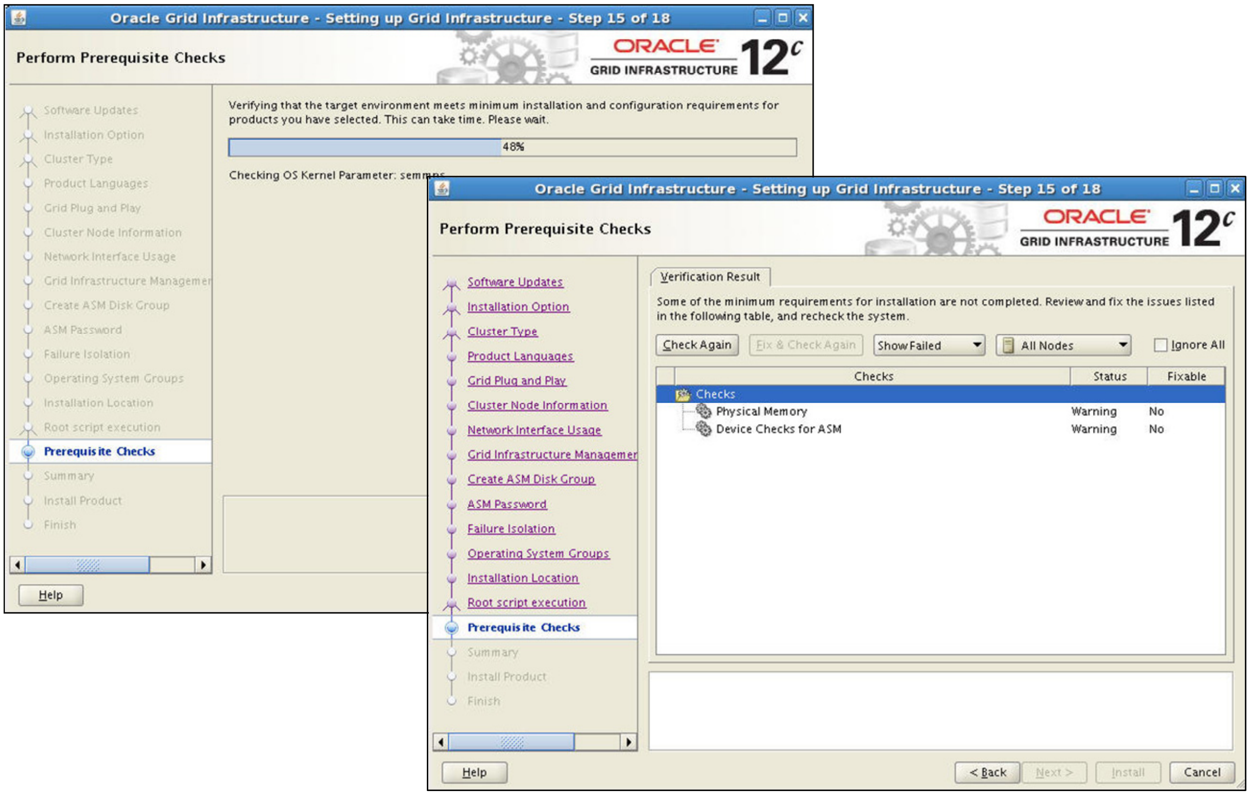
On the Create Inventory page, enter a location under your Oracle Base to store the Oracle Inventory or accept the default value. Click Next to continue.

Root Script Execution Configuration



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 don’t 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.

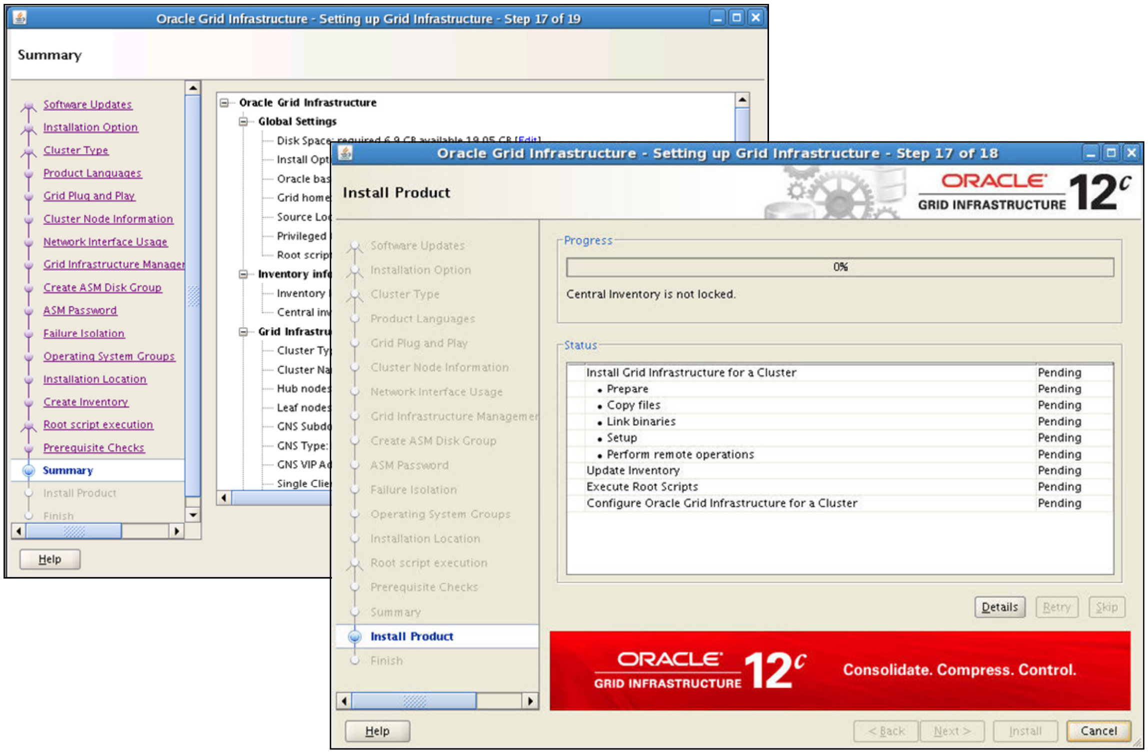
Perform Prerequisite Checks



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.

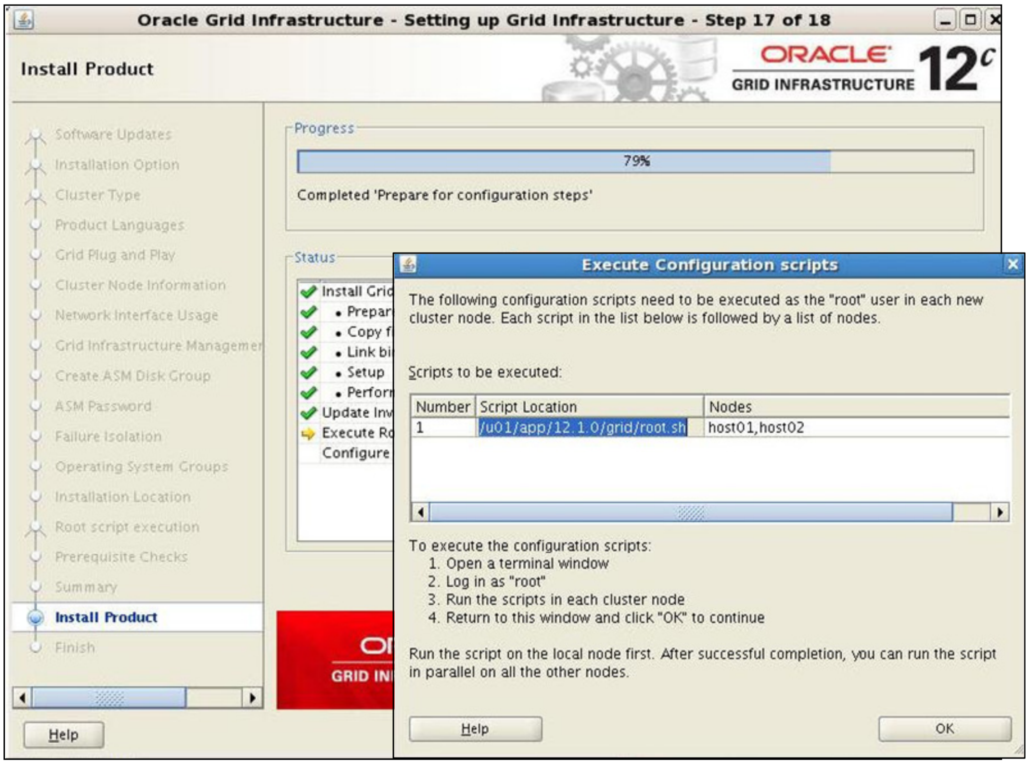
The installer can determine whether the deficiency can be corrected, presenting the user performing the installation with the option of allowing the OUI to correct the situation. By clicking the Fix & Check Again button, a script is generated on the nodes where the deficient condition exists. After executing 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



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

Root Scripts



If you chose to run the `root` scripts manually, you will be prompted to run the root scripts after Grid Infrastructure has been copied to the remote nodes. If this is the first Oracle software initiated by the operating system user, you will run the `orainstRoot.sh` scripts on all nodes as prompted by the installer. Next, you must execute the `root.sh` script on all nodes. When the `root.sh` script finishes executing on the last node, click Finish to exit the installer.

Verifying the Grid Infrastructure Installation

```
[root@host01 ~]# crsctl stat res -t
--------------------------------------------------------------------------------
Name           Target  State        Server                   State details
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.ASMNET1LSNR_ASM.lsnr
               ONLINE  ONLINE       host01                   STABLE
               ONLINE  ONLINE       host02                   STABLE
ora.DATA.dg
               ONLINE  ONLINE       host01                   STABLE
               ONLINE  ONLINE       host02                   STABLE
ora.LISTENER.lsnr
               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                               STABLE
ora.cvu
      1        ONLINE  ONLINE       host02                   STABLE
ora.gns
      1        ONLINE  ONLINE       host02                   STABLE
ora.gns.vip
      1        ONLINE  ONLINE       host02                   STABLE
...
```


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.

```
# 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.

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

- If you installed the OCR and voting disk files on ASM, check it with the `srvctl status asm` command:

```
# srvctl status asm
ASM is running on host01,host02
```

- ASM is running only if it is needed for Clusterware files.
- If you have not installed OCR and voting disks files on ASM, then the ASM instance should be down.

If you installed the OCR and voting disk files on Oracle ASM, then use the following command syntax as the Oracle Grid Infrastructure installation owner to confirm that your Oracle ASM installation is running:

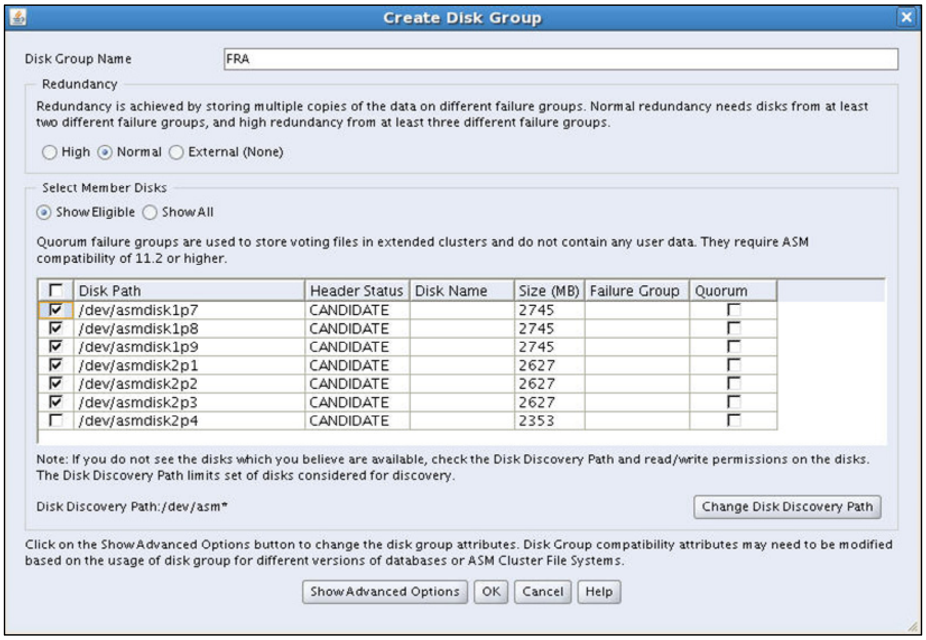
```
$ srvctl status asm
```

Oracle ASM will be running only if it is needed for Oracle Clusterware files. If you have not installed the OCR and voting disks files on Oracle ASM, the Oracle ASM instance should be down.

To manage Oracle ASM or Oracle Net 11*g* 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

If you plan to add an Oracle database, create a disk group for the Fast Recovery Area for database files by using ASMCA.



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/12.1.0/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/12.1.0/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.

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 path `Grid_home/crs/install`, where `Grid_home` is the path to the Grid home, and unlock the Grid home using the command `rootcrs.pl -unlock -crshome Grid_home`, where `Grid_home` is the path to your Grid infrastructure home. For example, with the Grid home `/u01/app/11.2.0/grid`, enter the following command:
   ```
   # cd /u01/app/12.1.0/grid/crs/install
   # perl rootcrs.pl -unlock -crshome /u01/app/12.1.0/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 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/12.1.0/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.

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/12.1.0/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

Answer: a

This statement is true.

Summary

In this lesson, you should have learned how to:

- Install Grid Infrastructure
- Verify the installation
- Configure Automatic Storage Management (ASM) disk groups

Practice 5: Overview

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

6

Managing Cluster Nodes

Objectives

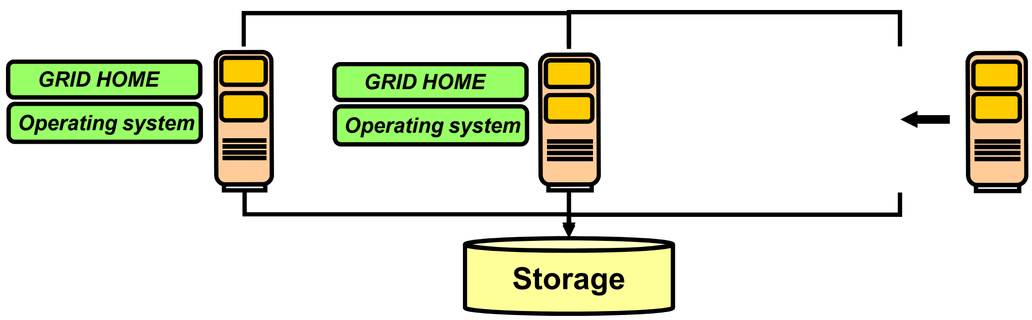
After completing this lesson, you should be able to:

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

Adding and Deleting Nodes

The `addNode.sh` shell script:

- Is used to add nodes to an existing Oracle Clusterware environment
- Runs without a graphical interface
- Does not perform the prerequisite operating system tasks



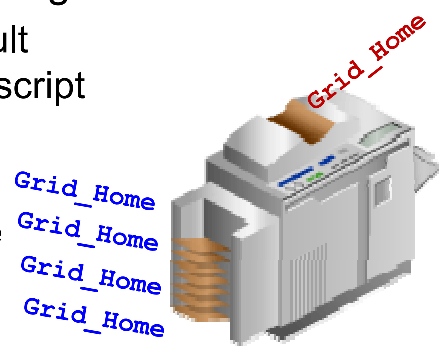
You can use a variety of methods to add and delete nodes in an Oracle Clusterware environment:

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

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

Cloning Oracle Clusterware

- Cloning is the process of copying an existing Clusterware installation to a different location and then updating the copied installation to work in the new environment.
- Cloning requires that you start with a successfully installed Oracle Grid Infrastructure home.
- This home is used as the basis for implementing a script that creates a cluster based on the original Grid home.
  - Creating the cloning script can result in errors because you prepare the script without interactive checks.
  - The initial effort is worthwhile for scenarios in which you run a single script to configure tens or even hundreds of clusters.



Cloning is the process of copying an existing Oracle Clusterware installation to a different location and then updating the copied installation to work in the new environment. Changes made by one-off patches applied on the source Oracle Grid Infrastructure home are also present after cloning. During cloning, you run a script that replays the actions that installed the Oracle Grid Infrastructure home.

Cloning requires that you start with a successfully installed Oracle Grid Infrastructure home. You use this home as the basis for implementing a script that extends the Oracle Grid Infrastructure home to create a cluster based on the original Grid home.

Manually creating the cloning script can be error prone because you prepare the script without interactive checks to validate your input. Despite this, the initial effort is worthwhile for scenarios where you run a single script to configure tens or even hundreds of clusters. If you have only one cluster to install, you should use the traditional, automated and interactive installation methods, such as Oracle Universal Installer (OUI) or the Provisioning Pack feature of Oracle Enterprise Manager.

Cloning is not a replacement for Oracle Enterprise Manager cloning that is a part of the Provisioning Pack. During Oracle Enterprise Manager cloning, the provisioning process simplifies cloning by interactively asking for details about the Oracle home. The interview questions cover such topics as the location to which you want to deploy the cloned environment, the name of the Oracle database home, a list of the nodes in the cluster, and so on.

The Provisioning Pack feature of Oracle Enterprise Manager Cloud Control provides a framework that automates the provisioning of nodes and clusters. For data centers with many clusters, the investment in creating a cloning procedure to provision new clusters and new nodes to existing clusters is worth the effort.

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:
   a. Check system requirements.
   b. Check network requirements.
   c. Install the required operating system packages.
   d. Set kernel parameters.
   e. Create groups and users.
   f. Create the required directories.
   g. Configure the installation owner's shell limits.
   h. Configure Secure Shell (SSH) and enable user equivalency.

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.
   a. Perform a post-hardware and operating system check.

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

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

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

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

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


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:

```
$ oclumon manage -get repsize
```

Add additional space, if required, as follows:

```
$ oclumon manage -repos changerepossize <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:

```
[grid@host01]$cd /<Grid_Home>/oui/bin
[grid@host01]$./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}"
```


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 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/12.1.0/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
```


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

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
```


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:

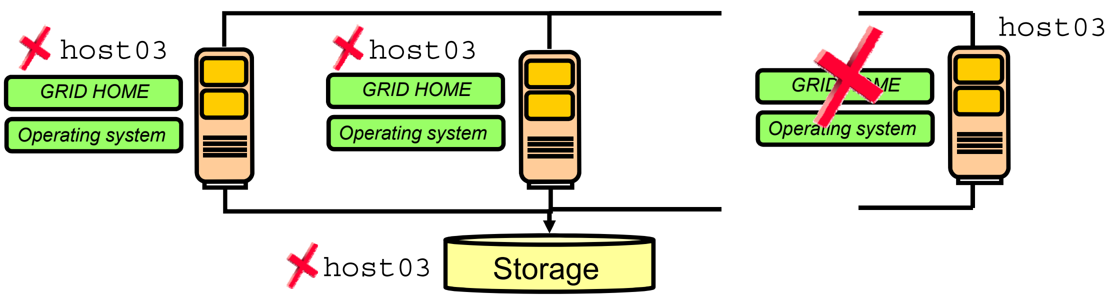
   ```
   C:\>cluvfy stage -post crsinst -n all [-verbose]
   ```

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.



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

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

Deleting a Node from the Cluster

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 `rootcrs.pl` script as `root` from the *`Grid_home`*`/crs/install` directory on each node that will be deleted:

```
[root@host03]# ./rootcrs.pl -delete -force
```


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

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

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

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

Deleting a Node from the Cluster

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

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

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

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


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

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

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

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

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

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

Deleting a Node from the Cluster

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

```
[grid@host03]$ ./runInstaller -detachHome
ORACLE_HOME=/u01/app/12.1.0/grid
```

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

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

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

```
[grid@host01]$ cd /Grid_home/oui/bin
[grid@host01]$ ./runInstaller -updateNodeList \
ORACLE_HOME=/u01/app/12.1.0/grid \
"CLUSTER_NODES={host01,host02}" CRS=TRUE
```

ORACLE

Step 6:

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

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

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

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

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

Deleting a Node from a Cluster (GNS in Use)

If your cluster uses GNS, perform *only* the following steps from the previous procedure:

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

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

Deleting a Node from the Cluster

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

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

ORACLE

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

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. On a remaining node, run the following command:

```
C:\>setup.exe -updateNodeList ORACLE_HOME=Grid_home
"CLUSTER_NODES={host01,host02}" CRS=TRUE -silent
```

4. Run this command from a node that you are not deleting:

```
C:\>Grid_home\bin\crsctl delete node -n host03
```

5. Verify that the node has been successfully deleted:

```
C:\>cluvfy stage -post nodedel -n node_list -verbose
```


To delete a node from a Windows-based cluster, follow the steps below:

1. Only if you have a local home, run the following command with `-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 any node that you are not deleting, run the following command from the *`Grid_home`*`\oui\bin` directory where *`remaining_nodes`*_list is a comma-delimited list of the nodes that are going to remain part of your cluster:

```
C:\>setup.exe -updateNodeList ORACLE_HOME=Grid_home
"CLUSTER_NODES={remaining_nodes_list}" CRS=TRUE -silent
```

4. On a node that you are not deleting, run the following command:

```
C:\>Grid_home\bin\crsctl delete node -n node_to_be_deleted
```

5. 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.

Quiz

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

a. True

b. False

Answer: b

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

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

Practice 6: Overview

In this practice, you extend your cluster to another Hub Node and another Leaf Node.

7

Traditional Clusterware Management

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

Managing Oracle Clusterware

Command-line utilities

- `crsctl` manages clusterware-related operations:
  - Starting and stopping Oracle Clusterware
  - Enabling and disabling Oracle Clusterware daemons
  - Registering cluster resources
- `srvctl` manages Oracle resource–related operations:
  - Starting and stopping database instances and services
- `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.

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

ORACLE

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 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::-
```


Use the `crsctl setperm` command to configure horizontal role separation 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 a specific node:

```
# crsctl start cluster
```

```
# crsctl stop cluster
```

- To start or stop OHAS on the local server:

```
# crsctl stop crs
```

```
# crsctl stop crs
```

- To enable or disable Clusterware on a specific node:

```
# crsctl enable crs
```

```
# crsctl disable crs
```


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
```

ORACLE

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/asmdisk1p1)  [DATA]
 2. ONLINE   4222108c7a504f0abf464cb75bb555e6 (/dev/asmdisk1p10) [DATA]
 3. ONLINE   ff5d5ccd2fa04f70bffba241b89de6be (/dev/asmdisk1p11) [DATA]
Located 3 voting disk(s).
```


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
```

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

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


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

```
$ grep voting ocssd.log
[    CSSD]2013-07-07 10:47:09.711 [100494224] >ERROR:
   clssnmvReadFatal: voting device corrupt
   (0x00000000/0x00000000/1//dev/sda6)
[    CSSD]2013-07-07 10:47:09.711 [3082128272] >ERROR:
   clssnmvReadFatal: voting device corrupt
   (0x00000000/0x00000000/2//dev/sda7)
```

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

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

Locating the OCR Automatic Backups

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

```
$ ocrconfig -showbackup auto
host02  2013/07/08 19:00:23 /u01/app/12.1.0/grid/cdata/cluster01/backup00.ocr
host02  2013/07/08 15:00:23 /u01/app/12.1.0/grid/cdata/cluster01/backup01.ocr
host02  2013/07/08 11:00:22 /u01/app/12.1.0/grid/cdata/cluster01/backup02.ocr
host02  2013/07/07 03:00:18 /u01/app/12.1.0/grid/cdata/cluster01/day.ocr
host02  2013/06/25 02:30:51 /u01/app/12.1.0/grid/cdata/cluster01/week.ocr
```

- 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.

ORACLE

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

The automatic backup is on a four-hour schedule, but only a limited number of files are retained. Only the last three backups of the routine four-hour intervals are kept, with newer backups overwriting older ones. At the end of the day, a backup is taken. 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.

Because the automatic backup is performed only by the master node to the local file system by default, it is recommended that you change the OCR automatic backup location to one that is shared by all nodes in the cluster by using the `ocrconfig -backuploc <new location>` command. The *file_name* variable can be a full directory path name that is accessible by all nodes, or it can be an Oracle ASM disk group that is mounted on all nodes. You must migrate OCR to Oracle ASM before changing the OCR backup location to an Oracle ASM disk group.

The location will be used for both automatic and manual backups. You cannot customize the backup frequencies, the number of retained copies, or the names of the backup files. If CRS on the master node is shut down, another node becomes the master, and backups resume on that node. If the backup location has not been changed to a common shared location, backups will exist locally across multiple nodes potentially. If CRS is stopped on all nodes during a scheduled backup, on restart, a backup is not immediately taken and the backup timer is reset. This could result in a longer time duration between automatic backups than the standard four-hour interval.

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

Adding, Replacing, and Repairing OCR Locations

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

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

- To replace the current OCR location:

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

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

```
# ocrconfig -repair -add +DATA1
```

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

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/sde1
```

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

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 the upgrade to Oracle Clusterware 12*c* is complete.

```
$ crsctl query crs activeversion
Oracle Clusterware active version on the cluster is [12.1.0.1.0]
```

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
```


To improve Oracle Clusterware storage manageability, OCR is configured, by default, to use ASM in Oracle Database 12*c*. With the Clusterware storage residing in an ASM disk group, you can manage both database and clusterware storage 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 12*c* 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 the upgrade to Oracle Clusterware 12*c* is complete.

```
$ crsctl query crs activeversion
Oracle Clusterware active version on the cluster is [12.1.0.1.0]
```

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

```
# ocrconfig -add /dev/sde2
# ocrconfig -add /dev/sde7
```

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

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


To migrate Oracle Clusterware storage from ASM to another storage choice, perform the following steps:

1. Ensure the upgrade to Oracle Clusterware 12*c* is complete. Run the following command to verify the current running version: `$ crsctl query crs activeversion`.
2. Create a shared file with the following permissions: `root`, `oinstall`, `640`, making sure that the mount partition has at least 300 MB of space.
3. To add the file as an OCR location, ensure that the Oracle Clusterware stack is running and run the following command as `root`:

   `# ocrconfig -add` *`new_file_location`*

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

   `# ocrconfig -delete` *`old_storage_location`*

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

Performing Manual OCR Backups

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
host02     2013/07/10 17:21:07
/u01/app/12.1.0/grid/cdata/cluster01/backup_20130710_172107.ocr
```


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

You are not allowed to specify the name used for the manual backup. A file will be created with the name `backup_<date>_<time>.ocr` and placed into the default backup location. When a manual backup is performed, it does not affect the automatic backup interval.

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

Restoring the OCR on Linux or UNIX Systems

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
```

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 the command below to start the Clusterware stack on one node in exclusive mode:

```
# crsctl start crs -excl -nocrs
```

5. 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.

ORACLE

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 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
   ```

If the preceding command returns any error due to OCR corruption, 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 command below to restore OCR with a backup identified using the `ocrconfig -showbackup` command.
   ```
   # ocrconfig -restore file_name
   ```

Restoring the OCR on Linux or UNIX Systems

6. Restore OCR with a backup that you can identify and verify its integrity as shown below:

```
# ocrconfig -restore /u01/app/.../cdata/cluster01/day.ocr
# 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
```

ORACLE

4. Run the command below 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`

5. If you want to restore OCR to an Oracle ASM disk group, you must first create a disk group 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;`

6. Restore OCR with a backup that you can identify by running the command below as `root`:
   ```
   # ocrconfig -restore file_name
   ```
   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 12*c*, 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 delete css votedisk path_to_voting_disk
# crsctl add css votedisk path_to_voting_disk
```


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}
```

ORACLE

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/asmdisk1p1)  [DATA]
 2. ONLINE  4222108c7a504f0abf464cb75bb555e6 (/dev/asmdisk1p10) [DATA]
 3. ONLINE  ff5d5ccd2fa04f70bffba241b89de6be (/dev/asmdisk1p11) [DATA]
Located 3 voting disk(s).
```


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/asmdisk1p1)  [DATA]
 2. ONLINE 4222108c7a504f0abf464cb75b555e6 (/dev/asmdisk1p10) [DATA]
 3. ONLINE ff5d5ccd2fa04f70bffba21b89de6be (/dev/asmdisk1p11) [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

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
```

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 FUID
```

Add a voting disk:

```
$ crsctl add css votedisk path_to_voting_disk
```

5. Stop the Oracle Clusterware stack:

```
# crsctl stop crs -f
```

6. Restart the Clusterware stack in normal mode:

```
# crsctl start crs
```


4. Depending on where you store your voting files, do one of the following:
   - 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.
   - 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
   ```

   Add a voting disk, as follows:

   ```
   $ crsctl add css votedisk path_to_voting_disk
   ```
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
```


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
eth1   192.168.1.0   global   cluster_interconnect,asm
```


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 -p -n
eth0  192.0.2.0   UNKNOWN   255.255.255.0
eth1  192.168.1.0  PRIVATE  255.255.255.0
eth1  169.254.0.0  UNKNOWN  255.255.0.0
eth2  192.168.1.0  PRIVATE  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:
```


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 with `oifcfg`

- Redundant Interconnect Usage can be configured after Clusterware installation.
- Use the **`oifcfg`** command to designate unused network interfaces as private.

```
$ oifcfg setif -global \
eth2/192.19.141.0: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
```


You can define multiple interfaces for Redundant Interconnect Usage after Clusterware installation by classifying the interfaces as private 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 using the `oifcfg setif` command, Clusterware must be stopped on all nodes and restarted using the `crsctl stop crs` and `crsctl start crs` commands, respectively.

Changing the Public VIP Addresses for Non-GPnP Clusters

1. Stop all services running on the node whose VIP address you want to change:

```
$ srvctl stop service -d orcl -s sales,oltp -n host01
```

2. Confirm the current IP address for the VIP address:

```
$ srvctl config vip -n host01
VIP exists: network number 1, hosting node host01
VIP IPv4 Address: -/host01-vip/192.0.2.237
VIP IPv6 Address:
```

3. Stop the VIP address:

```
$ srvctl stop vip -n host01
```

4. Verify that the VIP address is no longer running by executing the `ifconfig -a` command.

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

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

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

Changing the Public VIP Addresses for Non-GPnP Clusters

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

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

7. Start the node VIP.

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

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

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

ORACLE

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

Changing the Interconnect Adapter

- On a single node in the cluster, add the new global interface specification:

```
$ oifcfg setif -global eth2/192.0.2.0: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 eth1/192.168.1.0
# crsctl start crs
```


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

To change the network interface, perform the following steps:

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

7. Stop Oracle Clusterware by running the following command as `root` on each node:

   ```
   # crsctl stop crs
   ```

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
```

ORACLE

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]
```


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 `-scamname` 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
```

ORACLE

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 11*g* 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 12*c* is set to `SUBNET`.
- To change the valid node checking settings for the node listeners, edit the `listener.ora` file.

ORACLE

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 using the `srvctl`. 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 11*g* 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 12*c* 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 12*c* 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

Oracle Clusterware 12*c* 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 `srvctl` 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]
                   N            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]

--------------------------------------------------------------------------------
```


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 1 (12.1)*.

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` command or the `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]
                   Y            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]

--------------------------------------------------------------------------------
```


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 delete server my_server -f -admin

NAME = Free
ACTIVE_SERVERS  =

NAME = Generic
ACTIVE_SERVERS  =

NAME = ora.my_pool
ACTIVE_SERVERS  = some_server
```


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 } srvpool ... -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
```


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 1 (12.1).*

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
```


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 1 (12.1).*

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`

Answer: c

The correct answer is c.

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 using `ocrcheck`.

Answer: a, c, d

The correct answers are a, c, and d.

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

Answer: b

What-if command evaluation using the `crsctl eval ... resource` command is recommended for use only with user-defined application resources. For Oracle Clusterware resources (resources with the `ora.` name prefix), you should use the `srvctl predict` command or the `srvctl ... -eval` command.

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

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
- Performing what-if command evaluation by using the commands for various resources

8

Policy-Based Cluster Management

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

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 12*c*, 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”

ORACLE

Oracle Clusterware 11*g,* 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 12*c*, 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 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 11*g* Release 2

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 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.

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

ORACLE

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.

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.

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`.

Assume that there are no servers currently in a cluster, all server pools are empty, and server pools are defined according to the chart listed in the slide. When a server named `host01` joins the cluster, server-to-server pool assignment commences:

1. Oracle Clusterware processes only top-level server pools first (those that have no parent server pools). In this example, the top-level server pools are `sp1`, `sp2`, and `sp3`.
2. Oracle Clusterware lists the pools in order of `IMPORTANCE`, as follows: `sp2`, `sp3`, `sp1`.
3. Oracle Clusterware assigns `host01` to `sp2` because `sp2` has the highest `IMPORTANCE` value and its `MIN_SIZE` value has not yet been met.
4. Oracle Clusterware processes the remaining two server pools, `sp2_1` and `sp2_2`. The sizes of both server pools are below the value of the `MIN_SIZE` attribute (both server pools are empty and have `MIN_SIZE` values of `1)`.
5. Oracle Clusterware lists the two remaining pools in order of `IMPORTANCE`, as follows: `sp2_1`, `sp2_2`.
6. Oracle Clusterware assigns `host01 to sp2_1` but cannot assign `host01` to `sp2_2` because `sp2_1` is configured to be exclusive with `sp2_2`.

Creating Server Pools with `srvctl` and `crsctl`

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 srvpool -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
```

ORACLE

Use the `crsctl add serverpool` command or the `srvctl add srvpool` 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 -g 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
```

ORACLE

Use the `crsctl` utility or the `srvctl` utility to delete and modify server pools. Do not use the `crsctl` utility for any server pools with names that begin with `ora` because these server pools are Oracle server pools.

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.

If the number of servers in a server pool falls below the value of the `MIN_SIZE` attribute for the server pool (such as when a server fails), based on values you set for the `MIN_SIZE` and `IMPORTANCE` attributes for all server pools, Oracle Clusterware can move servers from other server pools into the server pool whose number of servers has fallen below the value for `MIN_SIZE`.

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*.

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

Oracle Clusterware assigns each server a set of attributes as soon as you add a server to a cluster. If you remove the server from the cluster, Oracle Clusterware revokes those settings. These attributes include:

NAME: The node name of the server. A server name can contain any platform-supported characters except the exclamation point (!) and the tilde (~). A server name cannot begin with a period or with ora. This attribute is required.

ACTIVE_POOLS: A space-delimited list of the names of the server pools to which a server belongs. Oracle Clusterware manages this list, automatically.

STATE: A server can be in one of the following states:

- **ONLINE:** The server is a member of the cluster and is available for resource placement.
- **OFFLINE:** The server is not currently a member of the cluster. Subsequently, it is not available for resource placement.
- **JOINING:** When a server joins a cluster, Oracle Clusterware processes it to ensure that it is valid for resource placement. Oracle Clusterware also checks the state of resources configured to run on the server. When the validity of the server and the state of the resources are determined, the server transitions out of the state.

- **`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.

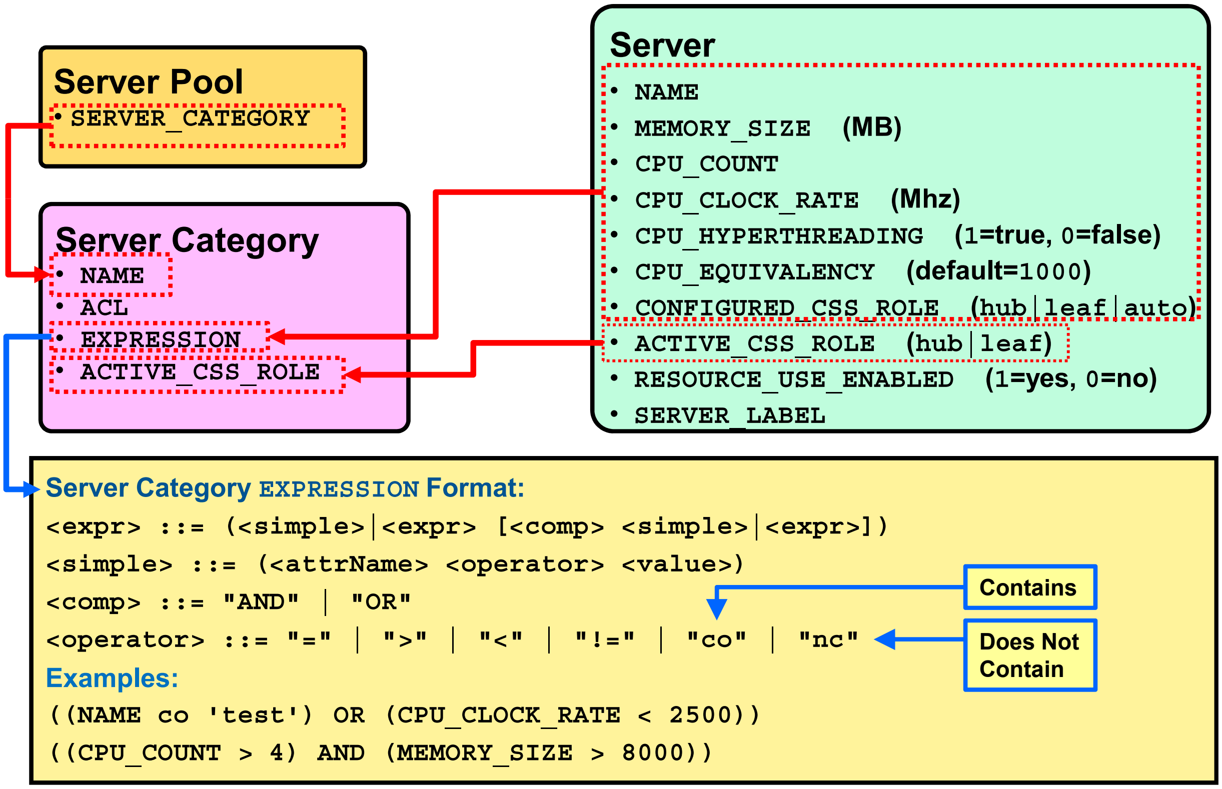
Server Categorization: Overview

- 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.

Oracle Clusterware 11*g* 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



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 12*c*, 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
```


Most server attribute settings are automatically discovered by Oracle Clusterware. An exception is the `SERVER_LABEL` attribute, which administrators can set by using the `crsctl modify server` command.

Administrators can view the server attributes by using the `crsctl status server` command. An example is displayed in the slide. Note the use of the `-f` option to display a full listing of all server attributes.

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>
```


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 for 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)
```

ORACLE

After a category is defined, it may be useful to understand which servers are members of that category. This can be achieved by using the `crsctl status server` command with the `-category` option, as shown in the slide.

It is also possible to list all of the categories that apply to a specific server by using the `crsctl status category` command with the `-server` option. Note that a server can belong to multiple categories at the same time, as shown in the example in the slide. The `ora.hub.category` category is an internal category that is used to categorize Hub Nodes.

Administering Server Categorization: Server Pools

- Specifying the `SERVER_CATEGORY` attribute:

```
$ crsctl add serverpool hr -attr "SERVER_CATEGORY='medium'" ...
```

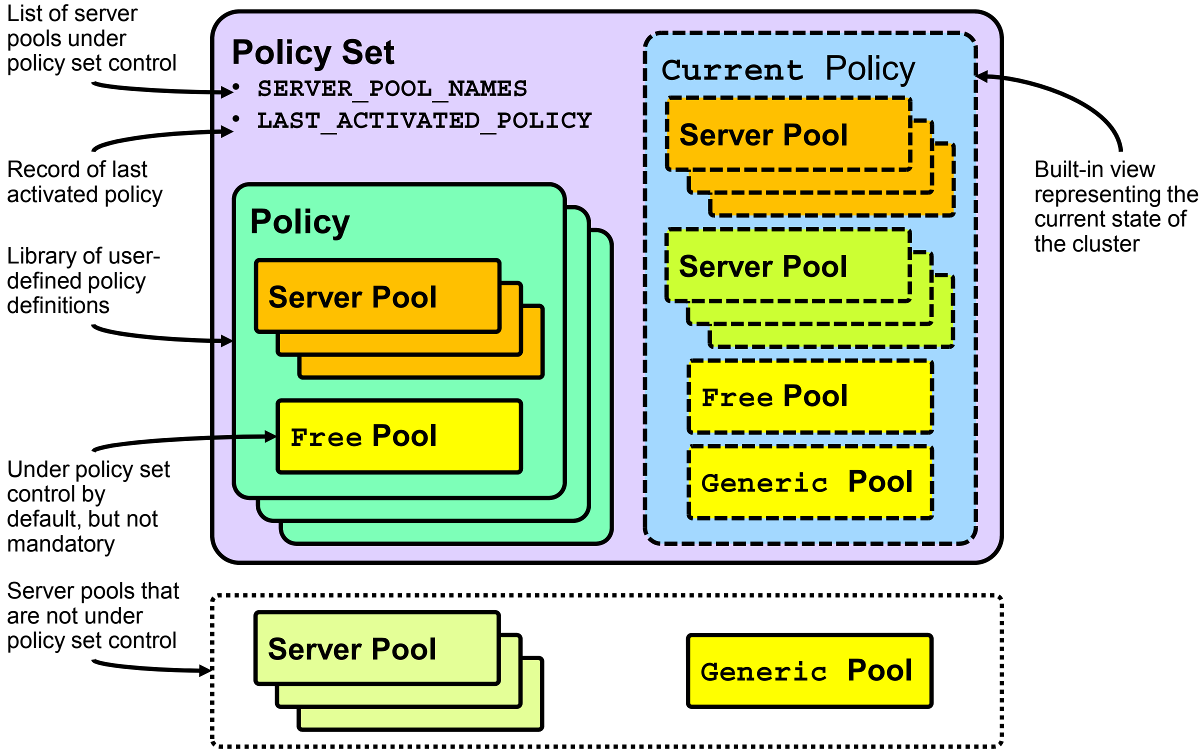
```
$ crsctl modify serverpool dev -attr "SERVER_CATEGORY='small'"
```

- 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
```


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 diagram in the slide illustrates the policy set contained in Oracle Clusterware 12*c*.

There is always exactly one policy set defined and used by the cluster. The policy set contains two attributes. The `SERVER_POOL_NAMES` attribute defines the names of all server pools controlled by the policy set. The policy set also contains an attribute that records the last activated policy.

The policy set contains zero or more user-defined policies. Each policy contains exactly one definition for each server pool controlled by the policy set. Typically, administrators create policies to address different priorities at different times.

Server pools may also exist outside the control of the policy set. For example, a server pool may be created with the `srvctl add serverpool` command and not be listed in the `SERVER_POOL_NAMES` policy set attribute. Server pools that are outside the control of the policy set are not directly affected by policy changes; however, such server pools may be indirectly affected by policy changes.

For example, a server pool outside the control of the policy set must yield a server when a policy change increases the number of servers allocated to a server pool with a higher priority, and there are no free servers or lower priority server pools elsewhere. Likewise, a server pool outside policy set control may grow in size because a policy change frees up a server.

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 12*c*, 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.

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 in order 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 12*c*. 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=
...
```


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
```


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
```


The example in this slide shows another way of configuring the policy set. It yields the same result as the commands for method 1.

This method of policy set configuration uses a text file that contains the policy set attribute definitions that are to be implemented. You can use the text file shown in this example as a template for configuring your own policy sets. In this example, the policy set contains two policies and two server pools; however, any number of policies and server pools can be specified this way. After the policy set definition file is created, the policy set can be modified by using the `crsctl modify policyset` command shown in the slide.

Administrators can also 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 edited and loaded back into the system by using the `crsctl modify policyset` command shown in the slide. This method is an effective way to configure the policy set when administrators start with existing server pool definitions that were created with previous clusterware releases.

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
```


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

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.

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.

Answer: a, d

A server category can contain no servers if none meets the criteria for entry into the category. Servers can belong to multiple categories simultaneously; however, they can belong only to one server pool at a time.

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.

Answer: b, e

There is always exactly one policy set defined in the cluster, and the policy set can contain zero or more policy definitions. The policy set automatically manages the server pools identified in the `SERVER_POOL_NAMES` attribute, and policy changes may indirectly affect other server pools.

Quiz

The `Current` policy __________ matches the policy definition specified in the `LAST_ACTIVATED_POLICY` policy set attribute.

a. Sometimes
b. Always
c. Never

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` 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

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

9

Upgrading and Patching Grid Infrastructure

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 patchset with the Oracle Universal Installer (OUI) utility
- Install a patch with the `opatch` utility

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 12*c* must have its own, new Grid home.
- For Oracle Clusterware 12*c*, 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.

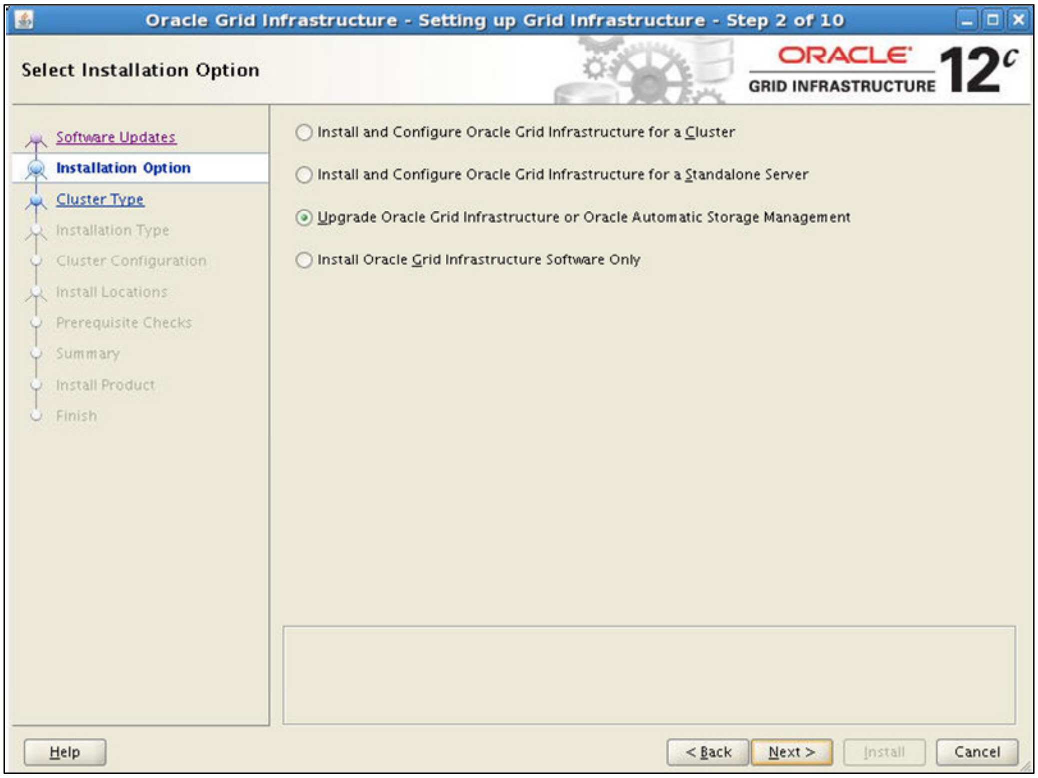
Oracle supports out-of-place upgrades, only, because Oracle Clusterware 12*c* must have its own, new Grid home. For Oracle Clusterware 12*c*, Oracle supports in-place or out-of-place patching. Oracle supports patch bundles and one-off patches for in-place patching but only supports patch sets and major 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 12*c*, 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 10*g* and Oracle Clusterware 11*g*; however, you can only upgrade Oracle ASM in a rolling manner from Oracle Database 11*g* release 1.

Oracle supports force upgrades in cases where some nodes of the cluster are down.

Oracle Clusterware Upgrade



Upgrading Oracle Clusterware using the Oracle Database 12*c* OUI allows you to perform an out-of-place upgrade. The Specify Installation Location page is displayed. You can enter a new directory under the current Oracle base in which to install the new Clusterware software, performing an out-of-place upgrade.

Options for Clusterware and ASM Upgrades

- Upgrade options from Oracle Grid Infrastructure 11*g* to Oracle Grid Infrastructure 12*c* include the following:
  - Rolling upgrades, with OCR and voting disks on ASM
  - Complete cluster (down time, non-rolling), with OCR and voting disks on ASM
  - Rolling upgrades with OCR and voting disks not on ASM
  - Complete cluster with OCR and voting disks not on ASM
- Upgrade options from releases *before* Grid Infrastructure 11.2 to Grid Infrastructure 12*c*, include the following:
  - Rolling upgrades with OCR and voting disks not on ASM
  - Complete cluster with OCR and voting disks not on ASM

Upgrade options from Oracle Grid Infrastructure 11*g* to Oracle Grid Infrastructure 12*c* include the following:

- Oracle Grid Infrastructure rolling upgrades, with OCR and voting disks on Oracle ASM
- Oracle Grid Infrastructure complete cluster (down time, nonrolling), with OCR and voting disks on Oracle ASM
- Oracle Grid Infrastructure rolling upgrades with OCR and voting disks on storage other than Oracle ASM
- Oracle Grid Infrastructure complete cluster (down time, nonrolling), with OCR and voting disks on storage other than Oracle ASM

Upgrade options from releases before Oracle Grid Infrastructure 11*g* Release 2 (11.2) to Oracle Grid Infrastructure 12*c* include the following:

- Oracle Grid Infrastructure rolling upgrades, with OCR and voting disks on storage other than Oracle ASM
- Oracle Grid Infrastructure complete cluster (down time, nonrolling), with OCR and voting disks on storage other than Oracle ASM

Preupgrade 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
```


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
```

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 `cluvfy.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 one below from the upgrade location:

```
$ ./runcluvfy.sh stage -pre crsinst -upgrade -n host01,host02
-rolling -src_crshome /u01/app/11.2.0/grid -dest_crshome
/u01/app/12.1.0/grid -dest_version 12.1.0.1 -fixup -verbose
```


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 command `runcluvfy.sh stage -pre crsinst -upgrade` 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 nodes host01 and host02 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/12.1.0/grid
-dest_version 12.1.0.1 -fixup -verbose
```

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.

ORACLE

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 11*g* 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.

Performing a Rolling Upgrade from an Earlier Release

1. Start the installer and select “Upgrade an existing Oracle Clusterware and Oracle ASM installation”.
2. On the node selection page, select all nodes.
3. Select installation options as prompted.
4. Run `root` scripts either automatically or manually:
   – If you configured `root` script automation, use the pause between batches to relocate services.
   – If you did not configure `root` script automation, run the `rootupgrade.sh` script on each node when prompted.

Use the following procedure to upgrade the cluster from an earlier release:

1. Start the installer, and select “Upgrade an existing Oracle Clusterware and Oracle ASM installation”.
2. On the node selection page, select all nodes.
3. Select installation options as prompted. It is recommended that you configure `root` script automation, so that the `rootupgrade.sh` script is run automatically during the upgrade.
4. Run `root` scripts 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.

Performing a Rolling Upgrade from an Earlier Release

5. If an earlier release of ASM is installed, the installer starts ASMCA to upgrade Oracle ASM to 12*c*.
   – You can choose to upgrade Oracle ASM at this time or upgrade it later.
6. Because the new Grid home is in a different location than the old Grid home, update any scripts or applications that use utilities, libraries, and so on residing in the Grid home.

5. After running the `rootupgrade.sh` script on the last node in the cluster, if you are upgrading from a release earlier than Oracle Grid Infrastructure 11*g* Release 2 and left the check box labeled ASMCA selected, as is the default, Oracle Automatic Storage Management Configuration Assistant (ASMCA) will run automatically, and the Oracle Grid Infrastructure upgrade will be complete. If you deselected the box during the interview stage of the upgrade, ASMCA is not run automatically.

If an earlier release of Oracle Automatic Storage Management (Oracle ASM) is installed, the installer starts ASMCA to upgrade Oracle ASM to 12*c*. You can choose to upgrade Oracle ASM at this time, or upgrade it later.

Oracle recommends that you upgrade Oracle ASM at the same time that you upgrade the Oracle Clusterware binaries. Until Oracle ASM is upgraded, Oracle Database installations that use Oracle ASM cannot be installed. Until Oracle ASM is upgraded, the 12*c* Oracle ASM management tools in the Grid home (for example, `srvctl`) do not work.

6. 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.

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/12.1.0/grid/rootupgrade -force
```

- Verify that the upgrade is complete by using the command `crsctl query crs activeversion` (as shown above).

ORACLE

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 command `crsctl query crs activeversion`. To resolve this problem, run the `rootupgrade.sh` command with the `-force` flag using the following syntax:

```
Grid_home/rootupgrade -force
```

Example:

```
# /u01/app/12.1.0/grid/rootupgrade -force
```

This command forces the upgrade to complete. Verify that the upgrade has completed by using the command `crsctl query crs activeversion`. 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.
- Inaccessible nodes have the following restriction after you run the `force` command:
  - After patch set upgrades, you can delete the node from the cluster. If the node becomes accessible later, and the patch release upgrade path is supported then you can upgrade it to the new patch release.

Unlocking the Old Oracle Clusterware Installation

1. As `root`, change the permission and ownership of the previous release Grid home:

```
# chmod -R 755 /u01/app/11.2.0/grid
# chown -R grid /u01/app/11.2.0/grid
# chown grid /u01/app/11.2.0
```

2. Log in as Installation owner and use the Grid Infrastructure 12*c* deinstallation tool to remove the old Grid home.
3. You can obtain the stand-alone deinstallation tool from the following URL: http://www.oracle.com/technetwork/database/enterprise-edition/downloads

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:

1. Log in as `root`, and change the permission and ownership of the previous release Grid home using the following command syntax, where `oldGH` is the previous release Grid home, *`swowner`* is the Oracle Grid Infrastructure installation owner, and *`oldGHParent`* is the parent directory of the previous release Grid home:
```
# chmod -R 755 oldGH
# chown -R swowner oldGH
# chown swowner oldGHParent
```
2. 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 12*c* stand-alone deinstallation tool to remove the previous release Grid home (*`oldGH`*).
3. 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.

Checking Cluster Health Monitor (CHM) Repository Size After Upgrading

- If you are upgrading from a prior release using IPD/OS, review the CHM repository size.
- It is recommended that you review your CHM repository needs and enlarge the repository size if you want to maintain a larger CHM repository.
- The default size is 1 GB or 1 hour (3600 seconds). To enlarge the CHM repository to 4 hours:

```
$ oclumon manage -repos changeretentiontime 14400
```

- If you enlarge the CHM repository size, be sure that there is local space available on each node of the cluster.
  - If there is insufficient space available, you can move the repository to shared storage.

If you are upgrading from a prior release using IPD/OS to Oracle Grid Infrastructure then review the Cluster Health Monitor (CHM) repository size. Oracle recommends that you review your CHM repository needs, and enlarge the repository size if you want to maintain a larger CHM repository.

By default, the CHM repository size is a minimum of either 1GB or 3600 seconds (1 hour). The CHM repository is one gigabyte, regardless of the size of the cluster. To enlarge the CHM repository, use the following command syntax, where *retention_time* is the size of CHM repository in number of seconds:

```
oclumon manage -repos changeretentiontime retention_time
```

The value for *retention_time* must be more than 3600 (one hour) and less than 259200 (three days). If you enlarge the CHM repository size, then you must ensure that there is local space available for the repository size you select on each node of the cluster. If there is not sufficient space available, then you can move the repository to shared storage.

For example, to set the repository size to four hours:

```
$ oclumon manage -repos changeretentiontime 14400
```

Note: The information here is pertinent only if a dedicated Grid Infrastructure Management Repository was not created.

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.

ORACLE

The patching cycle ensures that the bug fixes in the patch are applied appropriately.

Patching is unlike upgrading, which involves moving from one release of a product to another newer release of the software product.

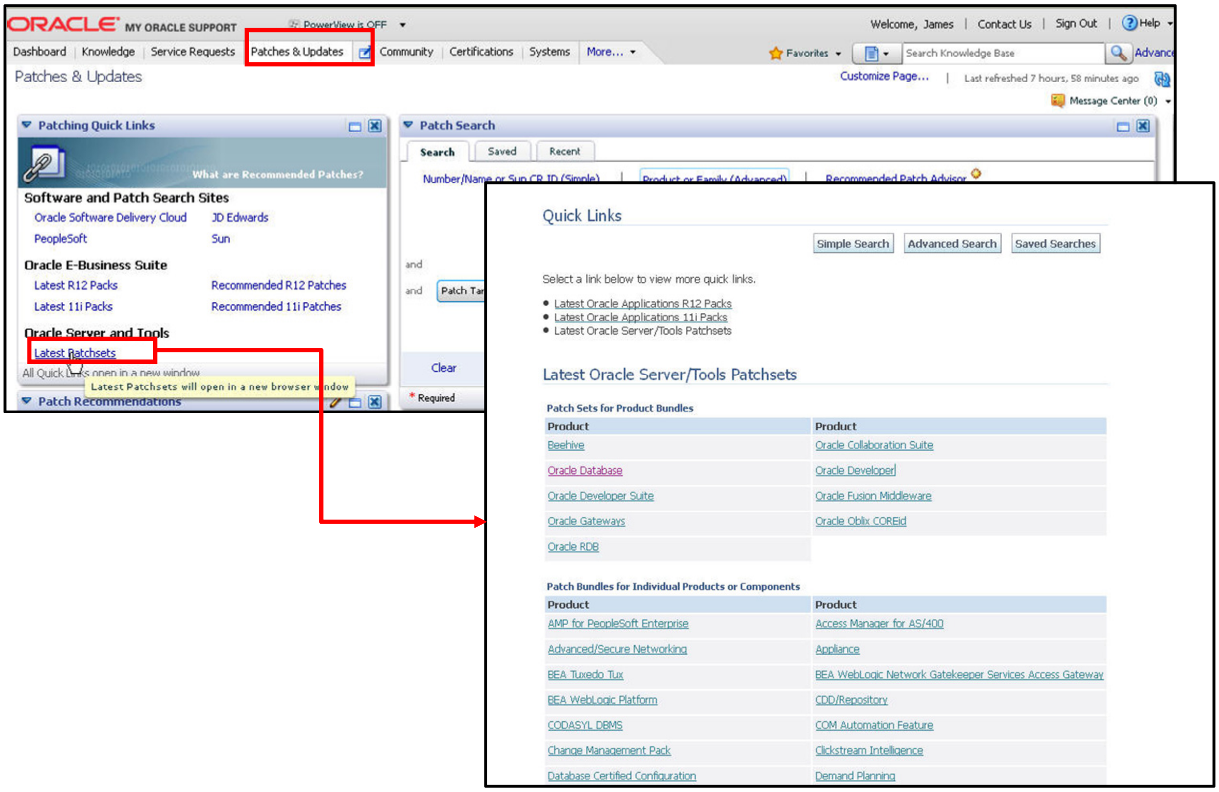
Types of Patches

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 bug fixes
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 recertification.
Security Patch Updates	A cumulative collection of security bug fixes; previously known as Critical Patch Updates (CPUs)

Interim patches are bug fixes available to customers in response to specific bugs. They require a particular base release or patchset to be installed before you can apply them. These patches are not versioned and are generally available in a future patchset 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, for example, from Oracle Database 11.2.0.3.0 to Oracle Database 11.2.0.3.2. You use OPatch to apply PSUs.

Obtaining Oracle Clusterware Patches



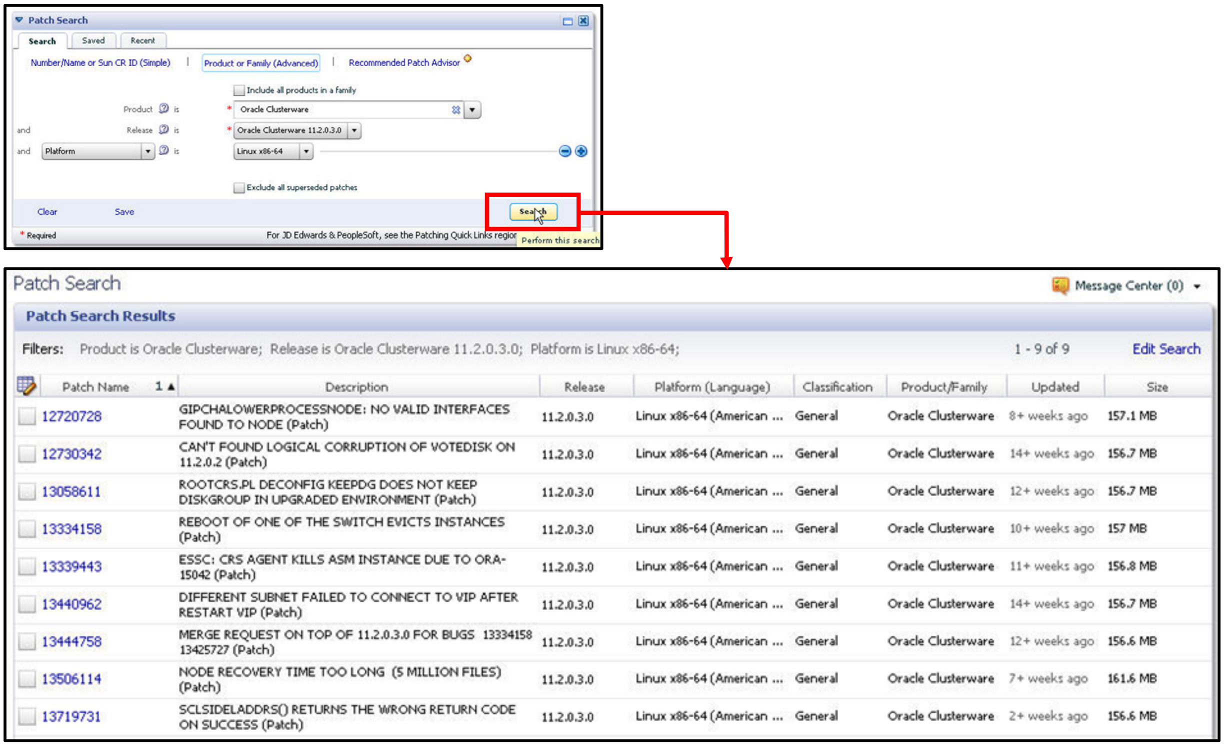
The latest patchsets and recommended CRS BPs can be downloaded from the My Oracle Support website at the following URL:

http://support.oracle.com/

After signing in to the website, click the “Patches & Updates” tab. For the latest patchsets and patch bundles, click the “Latest Patchsets” link under Oracle Servers and Tools. You can choose from patchsets for product bundles or patch bundles for individual products.

If you know the patchset number, you can click the Number/Name Sun CR ID link. You can enter a single patchset number or a comma-separated list. Select your platform and click the Search button on the Patch Search page.

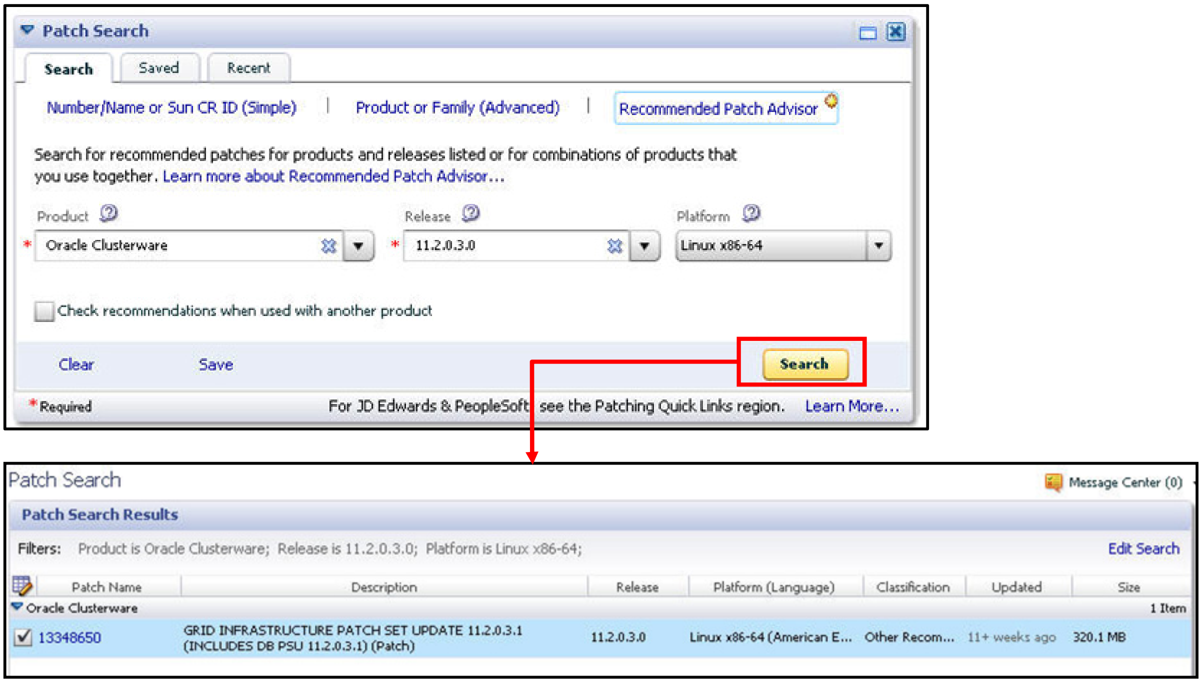
Obtaining Oracle Clusterware Patches



For a complete list of patchsets available for your Clusterware version and platform, click the Product or Family link.

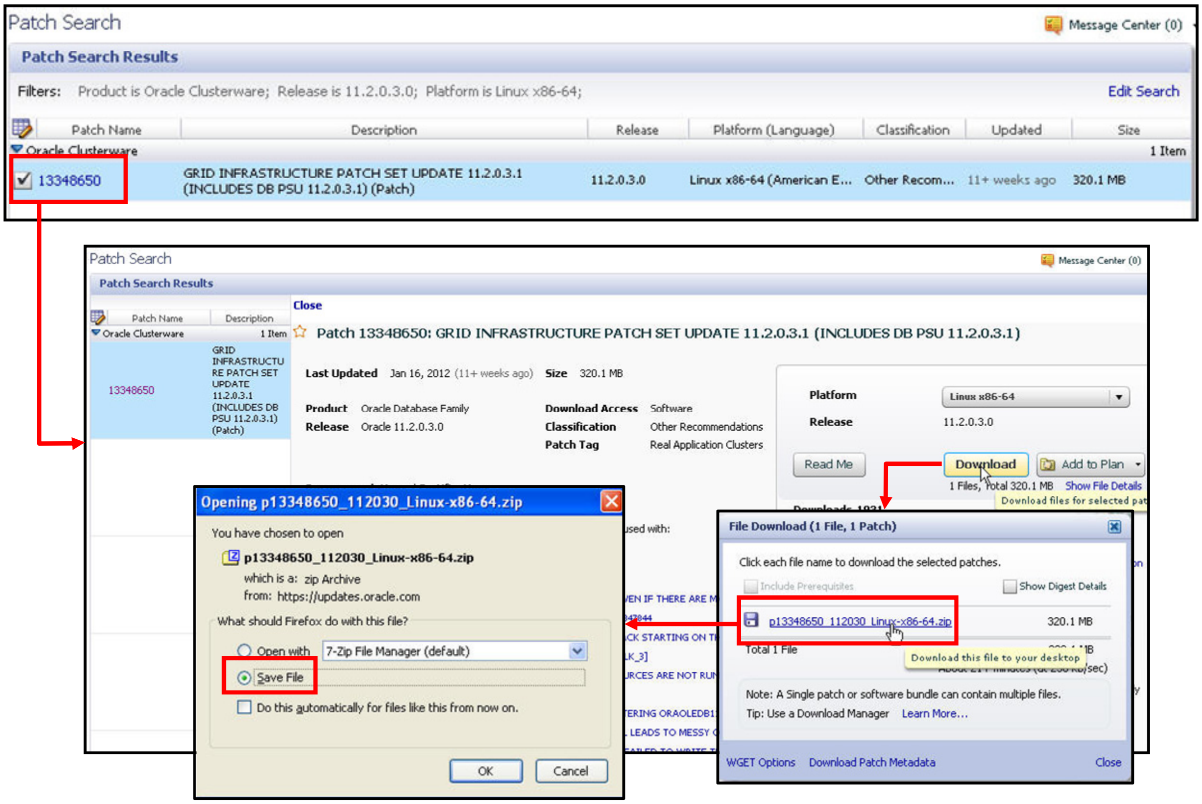
After specifying Product, Release, and Platform, click the Search button. The patch search results can then be viewed.

Obtaining Oracle Clusterware Patches



To locate recommended patchsets, start from the “Patches & Updates” tab and click the Recommended Patch Advisor link. Select Oracle Clusterware from the Product drop-down menu, and then select the release number and platform. Click the Search button for a list of recommended patchsets.

Downloading Patches



After 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 on 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.

Using EM Cloud Control for Patching Operations

- You can automate the patching of your Grid Infrastructure and RAC software with EM Cloud Control.
- Before Cloud Control can be used to patch Oracle software, you must perform the following configuration tasks:
  – Install the Enterprise Manager Agent on all cluster nodes.
  – Use PDP setup to ensure that the host user has the necessary privileges to complete patching tasks.
  – Configure named and preferred credentials in Enterprise Manager.
  – Configure a software library for storing patch files.
  – Download software from OTN to create installation media components in Enterprise Manager.

Using Cloud Control with its Provisioning & Patching functionality, you can automate the patching of your Oracle Grid Infrastructure and Oracle RAC software. Before you can use Cloud Control to patch your Oracle software, you must perform the following system configuration tasks:

- Install the Enterprise Manager Agent on all cluster nodes.
- Use PDP (Privilege Delegation Providers) setup to ensure the host user has the necessary privileges to complete patching tasks.
- Configure named and preferred credentials in Enterprise Manager.
- Configure a software library for storing patch files.
- Download software from OTN to create installation media components in Enterprise Manager.

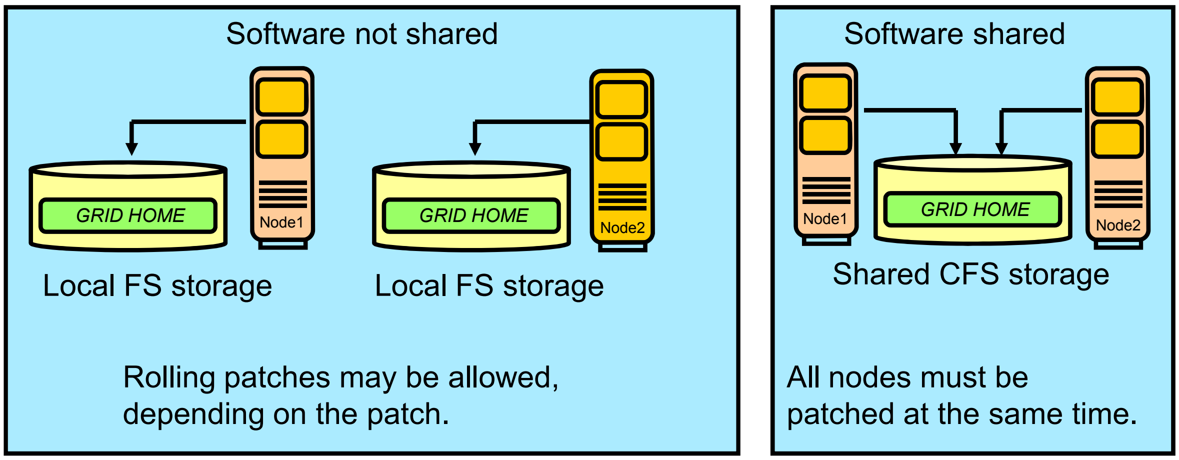
Details on how to perform these tasks, and how to patch your Oracle Grid Infrastructure and Oracle RAC software using EM Cloud Control are available from the following PDF file:

http://www.oracle.com/technetwork/oem/pdf/512066.pdf

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



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 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.

Minimum Down-Time Patching

- In minimum down-time 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 down time, compared to having all the nodes shut down at the same time.

In minimum down time 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 down time, compared to having all the nodes shut down at the same time.

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

- The local node is always patched first.
- The local node is used as a base to patch the other nodes.
- The user is prompted for the set of nodes to patch first from the remaining nodes.
- For each node in this first set, the user is asked to stop the instance and then the patch is propagated to that node before continuing to the next node. When the initial set of nodes has been patched, the user is asked to shut down the remaining nodes.

After the local node is patched, the patch is propagated to the last set of nodes and the inventory is updated. The last instances are stopped on the remote nodes. You can then start up the patched nodes (the first set of nodes) before patching the remaining nodes.

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.

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

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

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

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

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

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

Installing a Rolling Patchset with OUI

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

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

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

4. Start the newer OUI located from the patchset.

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


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

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

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

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

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

Installing a Rolling Patchset with OUI

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

5. Shut down the Oracle Clusterware and Oracle daemons.

```
# crsctl stop crs
```

6. Run the `root` upgrade script.

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

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

ORACLE

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

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

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

OPatch: Overview

- OPatch is a Java-based utility that allows the application and rolling back of interim patches.
  - 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
- A Grid home is partially owned by `root` and a grid owner, so the Grid home must be unlocked before patching:

```
# cd /u01/app/12.1.0/grid/crs/install
# perl rootcrs.pl -unlock -crshome /u01/app/12.1.0/grid
```

ORACLE

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 12*c* installation. When patching an Oracle Clusterware installation using the rolling-patch method, the user is prompted for which nodes to patch during the patch installation process. A rolling patch is identified inside the patch and cannot be enabled when invoking the patch tool. OPatch also supports a minimum down-time mode. An inventory of all patches that have been installed is maintained by OPatch.

Patching Oracle Clusterware has special considerations: Part of the software is owned by `root` and part is owned by the account used to install the product. Before you can apply a patch or make other modifications to the software files in the Grid home, you must first unlock the Grid home. To unlock the Grid home, perform the following steps:

1. Change directory to the path *`Grid_home`*`/crs/install`, where *`Grid_home`* is the path to the Grid home, and as the `root` user unlock the Grid home using commands similar to the following:
   ```
   cd /u01/app/12.1.0/grid/crs/install
   perl rootcrs.pl -unlock -crshome /u01/app/12.1.0/grid
   ```

When you run the `rootcrs.pl` script with the `-unlock` flag, it stops the Oracle Clusterware stack and unlocks the files in the Grid home so they can be modified.

2. Change user to the software owner for Oracle Grid Infrastructure for a cluster and apply the patch to the Grid home.
3. After you have finished modifying the Grid home, lock it again as the `root` user using commands similar to the following:

```
# cd /u01/app/12.1.0/grid/crs/install
# perl rootcrs.pl –patch
```

   The `rootcrs.pl` script with the `-patch` flag locks the Grid home again and restarts the Oracle Clusterware stack.

OPatch: General Usage

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

```
$ export ORACLE_HOME=/u01/app/12.1.0/grid
$ opatch command [options]
```

or

```
$ opatch command -oh /u01/app/12.1.0/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 -all | grep -i Rolling
```


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.

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 Rolling Patch with OPatch

1. Verify that Oracle Inventory is properly configured.

```
[grid]$ opatch lsinventory -detail -oh \
/u01/app/11.2.0/grid
```

2. Stop CRS on the first node to be patched.

```
[root]$ crsctl stop crs
```

3. Unlock the protected files.

```
[root]# cd <patch_location>
[root]# ./custom/scripts/prerootpatch.sh -crshome\
/u01/app/11.2.0/grid -crsuser grid
```

4. Save important configuration settings.

```
[grid]$ ./custom/scripts/prepatch.sh –crshome\
/u01/app/11.2.0/grid
```


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

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

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

```
crsctl stop crs
```

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

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

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

```
prepatch.sh –crshome Grid_home
```

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

Installing a Rolling Patch with OPatch

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

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

6. Apply configuration settings to the patched files.

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

7. Restore the lock to the protected files.

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

8. Verify patch installation.

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

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

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

```
opatch apply -local -oh grid_home patch_location
```

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

```
postpatch.sh -crshome Grid_home
```

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

```
postrootpatch.sh -crshome Grid_home
```

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

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

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

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

OPatch 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.

The OPatch utility has automated the patch application for the Oracle Grid Infrastructure home and the Oracle RAC database homes. It operates by querying existing configurations and automating the steps required for patching each Oracle RAC database home of the same version and the GI home.

The utility must be executed by an operating system user with root privileges (usually the user `root`), and it must be executed on each node in the cluster if the Grid Infrastructure home or Oracle RAC database home is in nonshared storage. The utility should not be run in parallel on the cluster nodes.

Depending on the command-line options specified, one invocation of OPatch can patch the GI home, one or more Oracle RAC database homes, or both GI and Oracle RAC database homes of the same Oracle release version. You can also roll back the patch with the same selectivity.

OPatch Automation: Examples

- 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>
```


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.

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

Using DBMS_QOPATCH, Oracle Database 12*c* 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

Quiz

Which tools can be used to install a patchset?

a. Oracle Universal Installer
b. OPatch
c. Enterprise Manager
d. Database Configuration Assistant

Answer: a, c

Oracle Universal Installer or Enterprise Manager can be used.

Quiz

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

a. True
b. False

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

10

Troubleshooting Oracle Clusterware

Objectives

After completing this lesson, you should be able to:

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

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

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

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/ntpd` file to add the `-x` flag, as in the following example:

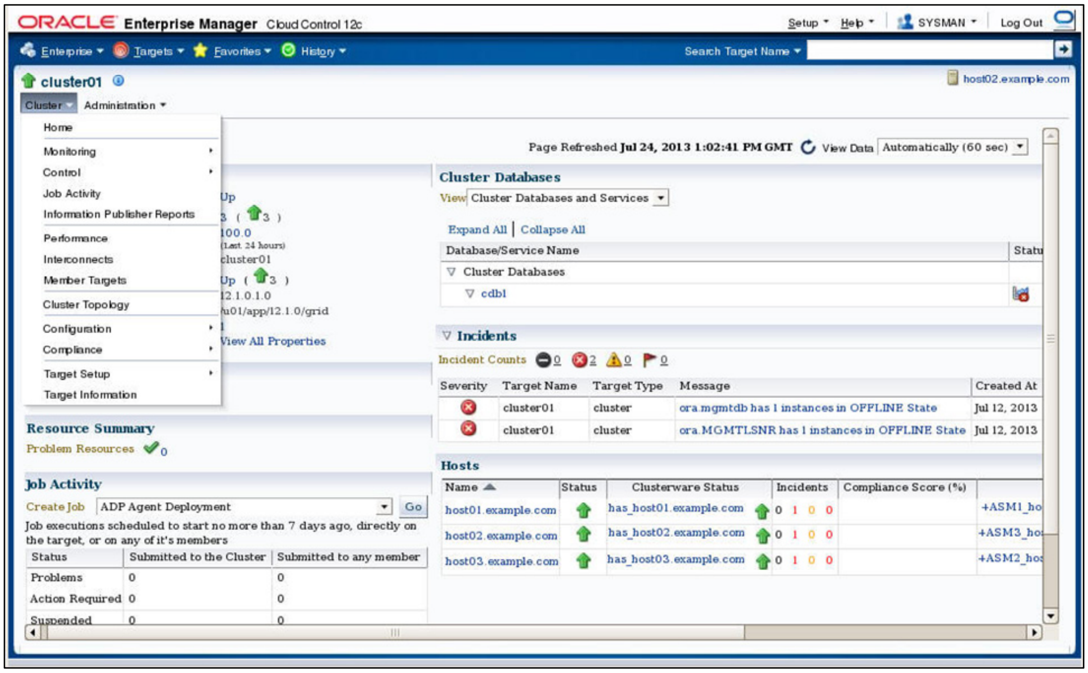
```
OPTIONS="-x -u ntp:ntp -p /var/run/ntpd.pid"
```

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

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

Monitoring Oracle Clusterware

You can use Oracle Enterprise Manager to monitor the Oracle Clusterware environment.



You can use Oracle Enterprise Manager to monitor the Oracle Clusterware environment. When you log in to Oracle Enterprise Manager using a client browser, the Cluster Database Home page appears where you can monitor the status of both Oracle Clusterware environments. Monitoring can include such things as:

- Notification of issues in the Oracle Clusterware alert log for the Oracle Cluster Registry, voting disk issues (if any), and node evictions
- Notification if there are any VIP relocations
- Status of the Oracle Clusterware on each node of the cluster using information obtained through the Cluster Verification Utility (`cluvfy`)
- Notification if node applications (nodeapps) start or stop

You can use the Oracle Enterprise Manager Interconnects page to monitor the Oracle Clusterware environment. The Interconnects page shows the public and private interfaces on the cluster, the overall throughput on the private interconnect, individual throughput on each of the network interfaces, error rates (if any), and the load contributed by database instances on the interconnect.

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 using the `oclumon` utility.

The Cluster Health Monitor (CHM) detects and analyzes operating system and cluster resource-related degradation and failures. CHM stores real-time operating system metrics in the Oracle Grid Infrastructure Management Repository that you can use for later triage with the help of My Oracle Support should you have cluster issues. CHM consists of the following services:

- System Monitor Service
- Cluster Logger Service
- Oracle Grid Infrastructure Management Repository

There is one system monitor service on every node. The system monitor service (`osysmond`) is the monitoring and operating system metric collection service that sends the data to the cluster logger service. The cluster logger service receives the information from all the nodes and persists in a Oracle Grid Infrastructure Management Repository database.

There is one cluster logger service, **`ologgerd`** for every 32 nodes in a cluster. Another **`ologgerd`** is spawned for every additional 32 nodes (which can be a sum of Hub and Leaf Nodes). If the cluster logger service fails (because the service is not able come up after a fixed number of retries or the node where it was running is down), Oracle Clusterware starts **`ologgerd`** on a different node. The cluster logger service manages the operating system metric database in the Oracle Grid Infrastructure Management Repository.

Grid Infrastructure Management Repository

- The Management Repository is an Oracle database that stores real-time metrics collected by CHM.
- The Management Repository is configured during an installation of or upgrade to Oracle Clusterware 12*c*.
- The Grid Infrastructure Repository:
  - Runs on one node in the cluster. Locate the repository on the same node as `ologgerd` to improve performance.
  - Communicates with any cluster clients such as `ologgerd` and `oclumon` through the private network
  - Communicates with external clients over the public network only
  - Data files are located in the same disk group as the OCR and voting disk

ORACLE

The Oracle Grid Infrastructure Management Repository is an Oracle database that stores real-time operating system metrics collected by CHM. You configure the Oracle Grid Infrastructure Management Repository during an installation of or upgrade to Oracle Clusterware 12*c* on a cluster.

The Grid Infrastructure Repository:

- Runs on one node in the cluster (this must be a Hub Node in an Oracle Flex Cluster configuration), and must support failover to another node in case of node or storage failure. You can locate the Oracle Grid Infrastructure Management Repository on the same node as the **`ologgerd`** to improve performance and decrease private network traffic.
- Communicates with any cluster clients such as OLOGGERD and OCLUMON through the private network. Oracle Grid Infrastructure Management Repository communicates with external clients over the public network only.
- Data files are located in the same disk group as the OCR and voting disk. Oracle increased the Oracle Clusterware shared storage requirement to accommodate the Oracle Grid Infrastructure Management Repository, which can be a network file system (NFS), cluster file system, or an Oracle ASM disk group.

`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`

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
```

`oclumon debug` Command

- The `oclumon debug` command is used to set the log level for CHM services.
- The syntax for the `oclumon debug` command is:

```
oclumon debug [log daemon module:log_level] [version]
```

- The following example sets the log level of the system monitor service (`osysmond`):

```
$ oclumon debug log osysmond CRFMOND:3
```

- To display daemon versions:

```
$  oclumon debug version
OCLUMON   version :0.02
OSYSMOND   version :0.23
OLOGGERD   version :2.01
NODEVIEW version :0.23
Clusterware version - label date: 12.1.0.1.0 - 130109
```


Use the `oclumon debug` command to set the log level for CHM services. The syntax for the `oclumon debug` command is:

`oclumon debug [log` *`daemon module`*`:`*`log_level`*`] [version]`

Supported daemon arguments for the command include: `osysmond`, `ologgerd`, `client`, and `all`. Supported *`daemon modules`* strings are:

- `osysmond: CRFMOND`, `CRFM`, and `allcomp`
- `ologgerd: CRFLOGD`, `CRFLDREP`, `CRFM`, and `allcomp`
- `client: OCLUMON`, `CRFM`, and `allcomp`
- `all: CRFM`, `allcomp`

Supported `log_level` values are 0, 1, 2, and 3. The `version` option is used to display the versions of the daemons.

`clumon dumpnodeview` Command

- The `oclumon 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 `oclumon dumpnodeview` command:

```
oclumon dumpnodeview [[-allnodes] | [-n node1 node2] [-last
"duration"] | [-s "time_stamp" -e "time_stamp"] [-v]
[-warning]] [-h]
```


Use the `oclumon 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"
```


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 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 `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 = 41097
 Done

$ oclumon manage -get reppath
CHM Repository Path =
+DATA/_MGMTDB/DATAFILE/sysmgmtdata.260.818370287

$ oclumon manage -get master
Master = host03
```


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.

Collecting CHM Data

- You can collect CHM data from any node in the cluster by running the `diagcollection.pl` script.
- The `diagcollection.pl` script creates a file called `chmosData_host_name_time_stamp.tar.gz.`
- To limit the amount of data you want collected, enter a command similar to the one below:

```
# <Grid_home>/bin/diagcollection.pl -collect -crshome
<Grid_home> -chmoshome <Grid_home> -chmos -incidenttime
12/21/2012 01:00:00 -incidentduration 00:30
```

Where `–incidenttime` is `MM/DD/YYYY24HH:MM:SS` and the format for `–incidentduration` is `HH:MM`.

You can collect CHM data from any node in the cluster by running the `diagcollection.pl` script on the node. To run the data collection script on only the node where the cluster logger service is running, perform the following steps:

1. Run the following command to identify the node running the cluster logger service:

   ```
   $ <Grid_home>/bin/oclumon manage -get master
   ```

2. Run the following command from a writable directory outside the Grid home as a privileged user on the cluster logger service node to collect all the available data in the Oracle Grid Infrastructure Management Repository:

   ```
   # <Grid_home>/bin/diagcollection.pl
   ```

The `diagcollection.pl` script creates a file called `chmosData_host_name_time_stamp.tar.gz`, similar to the following:

`chmosData_stact29_20121006_2321.tar.gz`

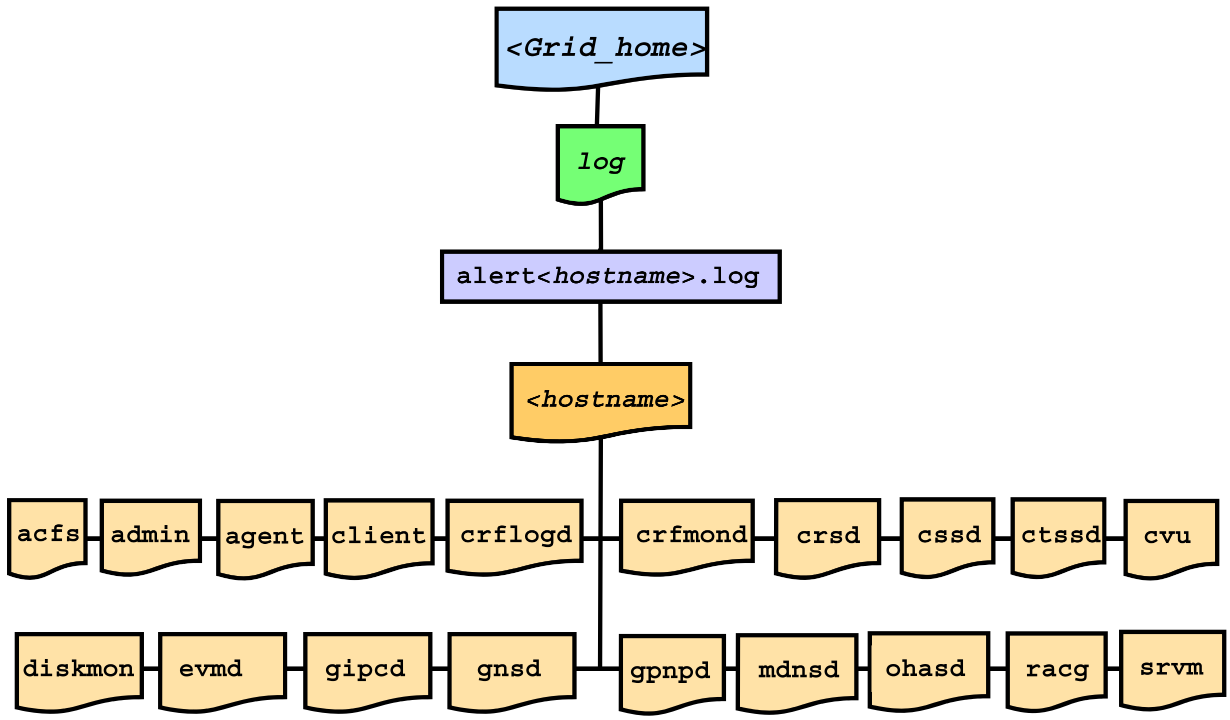
To limit the amount of data you want collected, enter the following command on a single line:

```
# <Grid_home>/bin/diagcollection.pl -collect -chmos
-incidenttime inc_time -incidentduration duration
```

In the preceding command, the format for the `-incidenttime` parameter is `MM/DD/YYYY24HH:MM:SS` and the format for the `-incidentduration` parameter is `HH:MM`. For example:

```
# <Grid_home>/bin/diagcollection.pl -collect -crshome
<Grid_home> -chmoshome <Grid_home> -chmos -incidenttime 12/21/2012
01:00:00 -incidentduration 00:30
```

Oracle Clusterware Main Log Files



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

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

- Important Oracle Clusterware alerts can be found in `alert<hostname>.log` in the `<Grid_home>/log/<hostname>` directory.
- The system monitor service and cluster logger service record log information in following locations, respectively: `<Grid_home>/log/host_name/crfmond` `<Grid_home>/log/host_name/crflogd`
- Cluster Ready Service (CRS) logs are in `<Grid_home>/log/<hostname>/crsd/`. The `crsd.log` file is archived for every 10 MB (`crsd.l01`, `crsd.l02`, …).
- Cluster Synchronization Service (CSS) logs are in `<Grid_home>/log/<hostname>/cssd/`. The `cssd.log` file is archived for every 20 MB (`cssd.l01`, `cssd.l02`, …).
- Event Manager (EVM) logs are in `<Grid_home>/log/<hostname>/evmd`.
- Grid client program such as `ocrdump`, `ocrconfig`, `ocrcheck`, `oifcfg`, and so on write logs in `<Grid_home>/log/<hostname>/client/` and `$ORACLE_HOME/log/<hostname>/client/`.
- Grid Naming Service logs can be found in `<Grid_home>/log/<hostname>/gnsd`.

The Oracle RAC high availability trace files are located in the following two locations:

`<Grid_home>/log/host_name/racg`, `$ORACLE_HOME/log/host_name/racg`

Oracle ACFS logs can be found in:

```
<Grid_home>/log/host_name/acfsrepl
<Grid_home>/log/host_name/acfsreplroot
<Grid_home>/log/host_name/acfssec
<Grid_home>/log/host_name/acfs
```

Oracle High Availability Services logs are found in `<Grid_home>/log/host_name/ohasd`.

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

Oracle Clusterware Alerts

- Oracle Clusterware posts alert messages when important events occur.
- Here is an example of an entry from the Grid Alert Log:

```
2013-06-17 20:33:46.746:
[ctssd(13395)]CRS-2403:The Cluster Time Synchronization Service on host
host01 is in observer mode.
2013-06-17 20:33:47.475:
[ctssd(13395)]CRS-2407:The new Cluster Time Synchronization Service
reference node is host host01.
2013-06-17 20:33:47.476:
[ctssd(13395)]CRS-2401:The Cluster Time Synchronization Service started
on host host01.
2013-06-17 20:39:24.180:
[crsd(15806)]CRS-1012:The OCR service started on node host01.
2013-06-17 20:39:25.157:
[crsd(15806)]CRS-1201:CRSD started on node host01.
[client(16172)]CRS-4742:OCR updated with contents of
/u01/app/12.1.0/grid/oc4j/j2ee/home/OC4J_DBWLM_config/system-jazn-
data.xml (New = 13378, Old = 0 bytes)
```


Oracle Clusterware posts alert messages when important events occur. The output in the slide is an example of an alert from the CRSD process. The location of this alert log on Linux, UNIX, and Windows systems is in the following directory path, where `<Grid_home>` is the name of the location where the Oracle Grid Infrastructure is installed: `<Grid_home>/log/host_name`.

The following example shows the start of the Oracle Cluster Time Synchronization Service (OCTSS) after a cluster reconfiguration:

```
2013-06-17 20:33:46.746:

[ctssd(13395)]CRS-2403:The Cluster Time Synchronization Service on
host host01 is in observer mode.

2013-06-17 20:33:47.475:

[ctssd(13395)]CRS-2407:The new Cluster Time Synchronization Service
reference node is host host01.

2013-06-17 20:33:47.476:

[ctssd(13395)]CRS-2401:The Cluster Time Synchronization Service
started on host host01.
```

Diagnostic Record Unique IDs (DRUIDs)

- Certain Oracle Clusterware messages contain a text identifier surrounded by "`(:`" and "`:)`".
  - Usually, the identifier is part of the message text that begins with "`Details in...`"
  - It includes a Clusterware diagnostic log file path and name.
- The identifier is called a Diagnostic Record Unique ID.

```
2013-06-17 20:42:09.546:
[/u01/app/12.1.0/grid/bin/orarootagent.bin(16308)]CRS-
5822:Agent '/u01/app/12.1.0/grid/bin/orarootagent_root'
disconnected from server. Details at (:CRSAGF00117:) {0:1:7} in
/u01/app/12.1.0/grid/log/host01/agent/crsd/orarootagent_root/or
arootagent_root.log.
```

- DRUIDs are used to relate external product messages to entries in a diagnostic log file and to internal Oracle Clusterware program code locations.

Certain Oracle Clusterware messages contain a text identifier surrounded by "`(:`" and "`:)`". Usually, the identifier is part of the message text that begins with "Details in..." and includes an Oracle Clusterware diagnostic log file path and name similar to the example in the slide. The identifier is called a DRUID, or Diagnostic Record Unique ID.

DRUIDs are used to relate external product messages to entries in a diagnostic log file and to internal Oracle Clusterware program code locations. They are not directly meaningful to customers and are used primarily by My Oracle Support when diagnosing problems.

Oracle Clusterware uses a file rotation approach for log files. If you cannot find the reference given in the file specified in the "Details in" section of an alert file message, this file might have been rolled over to a rollover version, typically ending in `*.l`*`number`* where *`number`* is a number that starts at 01 and increments to however many logs are being kept, the total for which can be different for different logs. Though there is usually no need to follow the reference unless you are asked to do so by My Oracle Support, you can check the path given for roll over versions of the file. The log retention policy, however, foresees that older logs are being purged as required by the amount of logs generated.

Diagnostics Collection Script

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

```
# /u01/app/12.1.0/grid/bin/diagcollection.pl --collect
Production Copyright 2004, 2010, Oracle.  All rights reserved

Cluster Ready Services (CRS) diagnostic collection tool
The following CRS diagnostic archives will be created in the local directory.
crsData_host01_20130715_1448.tar.gz -> logs,traces and cores from CRS home. Note:
core files will be packaged only with the --core option.

ocrData_host01_20130715_1448.tar.gz -> ocrdump, ocrcheck etc
coreData_host01_20130715_1448.tar.gz -> contents of CRS core files in text format
acfsData_host01_20130715_1448.tar.gz -> logs from acfs log.
osData_host01_20130715_1448.tar.gz -> logs from Operating System
...
```


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

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

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

Cluster Verify: Overview

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

ORACLE

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

There are two types of CVU commands:

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

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

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

Cluster Verify Components

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

```
$ cluvfy comp -list
```

ORACLE

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

nodereach	Checks node reachability	**peer**	Compares properties with peers
nodecon	Checks node connectivity	**ha**	Checks HA integrity
cfs	Checks CFS integrity	**asm**	Checks ASM integrity
ssa	Checks shared storage	**acfs**	Checks ACFS integrity
space	Checks space availability	**olr**	Checks OLR integrity
sys	Checks minimum requirements	**gpnp**	Checks GPnP integrity
clu	Checks cluster integrity	**gns**	Checks GNS integrity
clumgr	Checks cluster manager integrity	**scan**	Checks SCAN configuration
ocr	Checks OCR integrity	**ohasd**	Checks OHASD integrity
admprv	Checks administrative privileges	**crs**	Checks CRS integrity
software	Checks software distribution	**vdisk**	Checks Voting Disk Udev settings
clocksync	Checks clock synchronization	**dhcp**	Checks DHCP configuration
nodeapp	Checks node app existence	**dns**	Checks DNS configuration
freespace	Checks free space in CRS Home	**baseline**	Collects and compares baselines
healthcheck	Checks mandatory requirements		

Cluster Verify Locations

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

Cluster Verification Utility (CVU) was first released in Oracle Clusterware Release 10.2.0.1.0. CVU supports 12*c*, 11*g*R2, 11*g*R1, and 10*g*R2, as well as 10*g*R1 for Oracle Clusterware and RAC products. CVU is available in three different forms:

- Available on Oracle Technology Network (OTN) at: http://www.oracle.com/technetwork/products/clustering/downloads/index.html
  Click the Cluster Verification Utility (CVU) Download link. From there, you need to download the package for your platform and unzip it to a local directory (`<cvhome>`). You can use the `cluvfy` command from `<cvhome>/bin`. Optionally, you can set the `CV_DESTLOC` environment variable. This should point to a writable area on all nodes. CVU attempts to copy the necessary bits as required to this location. If this variable is not set, CVU uses `/tmp` as the default.
- Available in 12*c* distribution Oracle software DVD as a packaged version. Make use of `runcluvfy.sh`, which is needed when nothing is installed.
- Installed in both 12.1 Oracle Clusterware and RAC homes. Make use of `cluvfy` if the CRS software stack is installed. If the CRS software is installed, you can find `cluvfy` under `<Grid_home>/bin`.

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

Cluster Verify Configuration File

```
$ cat cvu_config
# Configuration file for Cluster Verification Utility(CVU)
# Version: 011405
#
#If CRS home is not installed, this list will be
#picked up when -n all is mentioned in the commandline argument.
#CV_NODE_ALL=

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

# Fallback to this distribution id
CV_ASSUME_DISTID=OEL4

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

# To override SSH location
#ORACLE_SRVM_REMOTESHELL=/usr/bin/ssh
...
# To override version used by command line parser
CV_ASSUME_CL_VERSION=12.1
```


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

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

- **`CV_ASSUME_CL_VERSION`:** To override version used by command-line parser
- **`CV_SUDO_BINARY_LOCATION`:** Complete file system path of `sudo` binary file, default is `/usr/local/bin/sudo`
- **`CV_PBRUN_BINARY_LOCATION`:** Complete file system path of `pbrun` binary file. The `pbrun` command is used to run a command or application in a controlled account.
- **`ORACLE_SRVM_REMOTESHELL`:** If set, this specifies the location for the `ssh`/`rsh` command to override CVU's default value. By default, this entry is commented out and the tool uses `/usr/sbin/ssh` and `/usr/sbin/rsh`.
- **`ORACLE_SRVM_REMOTECOPY`:** If set, this specifies the location for the `scp` or `rcp` command to override the CVU default value. By default, this entry is commented out and CVU uses `/usr/bin/scp` and `/usr/sbin/rcp`.

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

Cluster Verify Output: Example

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

Verifying CRS integrity

Checking CRS integrity...

Clusterware version consistency passed.
The Oracle clusterware is healthy on node "host03"
The Oracle clusterware is healthy on node "host02"
The Oracle clusterware is healthy on node "host01"

CRS integrity check passed

Verification of CRS integrity was successful.
```


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

Enabling Resource Debugging

- Change the `_USR_ORA_DEBUG` resource attribute to `1` for specific resources using `crsctl set log` command:

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

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

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

  – Or edit `<Grid_home>/bin/racgvip`:

```
vi /u01/app/12.1.0/grid/bin/racgvip
# Uncomment out the following line to enable debug tracing
# _USR_ORA_DEBUG=1
```

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

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

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

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

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

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

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

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

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

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

You can also use an initialization file when using OCR utilities. The initialization file for `ocrconfig`, `ocrdump`, and `ocrcheck` is:

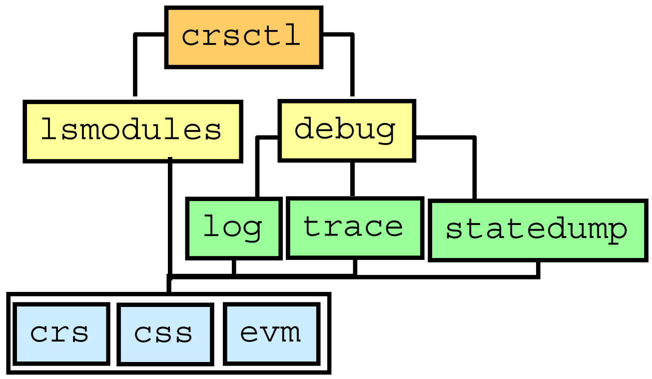
`Grid_home/srvm/admin/ocrlog.ini`

The format of the initialization file is as follows:

```
mesg_logging_level=2
comploglvl="OCRAPI:0;OCRSRV:0"
comptrclvl="OCRAPI:0;OCRSRV:0"
```

You can specify component names, logging, and tracing levels.

Dynamic Debugging




```
# crsctl lsmodules css
List CSSD Debug Module: BCCM         List CSSD Debug Module: GIPCCM
List CSSD Debug Module: CLSF         List CSSD Debug Module: GIPCGM
List CSSD Debug Module: CLSFA        List CSSD Debug Module: GIPCNM
List CSSD Debug Module: CLSINET      List CSSD Debug Module: GPNP
List CSSD Debug Module: CSSD         List CSSD Debug Module: OLR
List CSSD Debug Module: GIPCBCCM     List CSSD Debug Module: SKGFD
```

```
crsctl set log {crs | css | evm} "component_name=debugging_level,[...]"
```

```
# crsctl set log crs "CRSEVT=1","CRSCOMM=2","CRSTIMER=5"
```


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

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

The example in the slide shows you how to dynamically enable additional logging for the following CRS components: CRSEVT, CRSTIMER, and CRSCOMM.

Components for CRS, CSS, and EVM Modules

CRS Components		CSS Components	EVM Components
AGENT	CRSRES	BCCM	CLSVER
AGFW	CRSRPT	CLSF	CLUCLS
CLSEVT	CRSRTI	CLSFA	COMMCRS
CLSFRAME	CRSSE	CLSINET	COMMNS
CLSINET	CRSSEC	CSSD	CRSCCL
CLSO	CRSTIMER	GIPCBCCM	CRSOCR
CLSVER	CRSUI	GIPCCM	CSSCLNT
CLUCLS	CSSCLNT	GIPCGM	EVMAGENT
COMMCRS	OCRAPI	GIPCNM	EVMAPP
COMMNS	OCRASM	GPNP	EVMCOMM
CRSAPP	OCRCAC	OLR	EVMD
CRSCCL	OCRCLI	SKGFD	EVMEVT
CRSCEVT	OCRMAS		EVMMAIN
CRSCOMM	OCRMSG		OCRAPI
CRSD	OCROSD		OCRCLI
CRSEVT	OCRRAW		OCRMSG
CRSMAIN	OCRSRV		
CRSOCR	OCRUTL		
CRSPE	SuiteTes		
CRSPLACE	UiServer		

The table in the slide shows the components for the CRS, CSS, and EVM modules, respectively. Note that some component names are common between the CRS, EVM, and CSS daemons and may be enabled on that specific daemon. For example, `COMMNS` is the NS layer, and because each daemon uses the NS layer, you can enable this specific module component on any of the daemons to get specific debugging information.

CRS Component Descriptions

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

The table in the slide describes the functions performed by each CRS component listed.

OCR Component Descriptions

Module	Description
OCRAPI	CSS client component
OCRCLI	CSS daemon component
OCRSRV	OCR server component
OCRMAS	OCR master thread component
OCRMSG	OCR message component
OCRCAC	OCR cache component
OCRRAW	OCR raw device component
OCRUTL	OCR util component
OCROSD	OCR operating system dependent (OSD) layer
OCRASM	OCR ASM component
CRCONFIG	OCRCONFIG component for configuring OCR
OCRDUMP	OCRDUMP component that lists the OCR contents
OCRCHECK	OCRCHECK component that verifies all of the configured OCRs

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

CSS and COMM Component Descriptions

CSS:

Module	Description
`CSS`	CSS client component
`CSSD`	CSS daemon component

COMM:

Module	Description
`COMMCRS`	Clusterware communication layer
`COMMNS`	NS communication layer

The table in the slide describes the functions performed by CSS and COMM components.

Enabling Tracing for Java-Based Tools

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

```
$ export SRVM_TRACE=TRUE
$ srvctl config database -d orcl > /tmp/srvctl.trc
$ less  /tmp/srvctl.trc
...
[main] [ 2013-07-15 11:48:14.484 UTC ]
[OPSCTLDriver.setInternalDebugLevel:227]  tracing is true at level 2
to file null
[main] [ 2013-07-15 11:48:14.491 UTC ] [OPSCTLDriver.main:156]
SRVCTL arguments : args[0]=config args[1]=database args[2]=.d
args[3]=orcl
[main] [ 2013-07-15 11:48:14.532 UTC ] [Version.isPre:481]  version
to be checked 12.1.0.1.0 major version to check against10
[main] [ 2013-07-15 11:48:14.535 UTC ] [OCR.loadLibrary:312]
 Inside constructor of OCR
[main] [ 2013-07-15 11:48:14.548 UTC ] [OCR.loadLibrary:320]  Going
to load the ocr library
...
```


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

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

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

```
export SRVM_TRACE=
```

Preserving Log Files Before Wrapping

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

```
# Script to archive log files before wrapping occurs
# Written for CSS logs. Modify for other log file types.
CSSLOGDIR=/u01/app/12.1.0/grid/log/host01/cssd
while [ 1 -ne 0 ]; do
   CSSFILE=/tmp/css_`date +%m%d%y"_"%H%M`.tar
   tar -cf $CSSFILE $CSSLOGDIR/*
   sleep 300
done
exit
```

- Execute the script created in the background.

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


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

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

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 restart*, an eviction may just cause a restart of the Clusterware stack without rebooting the node.
- The following processes can evict nodes from the cluster or cause a node reboot:
  – `oclskd`: Is used by CSS to reboot a node based on requests from other nodes in the cluster
  – `cssdagent` and `cssdmonitor`: Monitor node hangs and vendor clusterware
  – `ocssd`: Internode health monitoring

ORACLE

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.

CSS uses the Oracle Clusterware Kill Daemon (`oclskd`) to stop processes associated with CSS group members for which stop requests have come. The `cssdagent` process is spawned by `ohasd` and is responsible for spawning the `ocssd` process, monitoring 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. OCSSD's primary job is 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 `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.

Determining Which Process Caused Reboot

Log file locations for processes causing reboots:

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

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

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

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

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

Reboot Advisory

- 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.

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 the 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

- 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
```


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 doesn’t 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 timestamp (approximately the time of the reboot). An example looks like this:

```
[ohasd(24687)]CRS-8011:reboot advisory message from host: sta00129,
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.

Using `ocrdump` to View Logical Contents of the OCR

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

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

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

- To dump the OCR contents for a specific key:

```
# ocrdump -keyname SYSTEM.css
```

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

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

- To dump the contents of an OCR backup file:

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

ORACLE

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

As `root`, `ocrdump -stdout | wc -l` results in 2405 lines on a test system, and

as `grid`, `ocrdump -stdout | wc -l` results in 1220 lines on the same system.

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

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

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

Checking the Integrity of the OCR

Use the `ocrcheck` command to check OCR integrity.

```
# ocrcheck
Status of Oracle Cluster Registry is as follows :
         Version                  :          4
         Total space (kbytes)     :     409568
         Used space (kbytes)      :       1892
         Available space (kbytes) :     407676
         ID                       : 1606747205
         Device/File Name         :      +DATA
                         Device/File integrity check succeeded

                                    Device/File not configured
                                    ...
         Cluster registry integrity check succeeded

         Logical corruption check succeeded
```


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

OCR-Related Tools for Debugging

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

```
mesg_logging_level = 5
comploglvl="OCRAPI:5 ; OCRSRV:5; OCRCAC:5; OCRMAS:5; OCRCLI:5; OCROSD:5"
comptrclvl="OCRAPI:5 ; OCRSRV:5; OCRCAC:5; OCRMAS:5; OCRCLI:5; OCROSD:5"
```


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

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

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

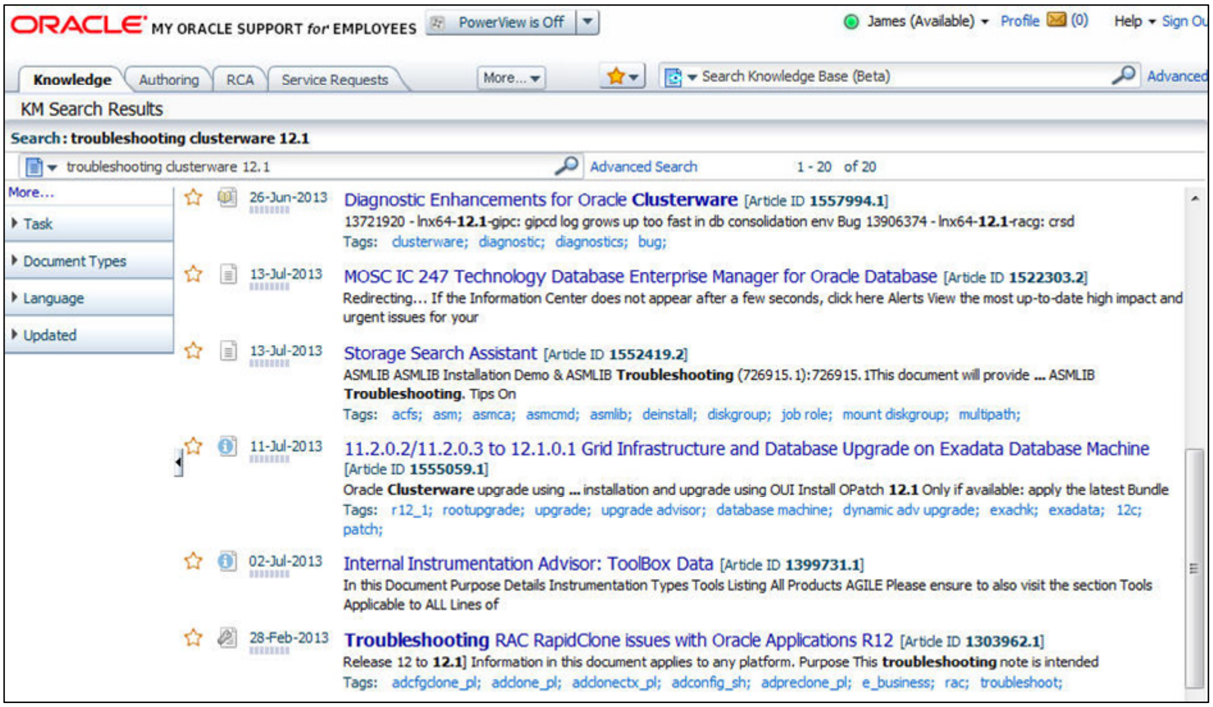
The slide shows you the three lines you could add to `ocrlog.ini` to turn on additional debugging information for some OCR components. A typical example where you may have to change the `ocrlog.ini` file is in a situation where you get errors while using either the `ocrdump` or `ocrconfig` tool.

OCR-Related Tools for Debugging

Module	Description
OCRAPI	CSS client component
OCRCLI	CSS daemon component
OCRSRV	OCR server component
OCRMAS	OCR master thread component
OCRMSG	OCR message component
OCRCAC	OCR cache component
OCRRAW	OCR raw device component
OCRUTL	OCR util component
OCROSD	OCR operating system dependent (OSD) layer
OCRASM	OCR ASM component
CRCONFIG	OCRCONFIG component for configuring OCR
OCRDUMP	OCRDUMP component that lists the OCR contents
OCRCHECK	OCRCHECK component that verifies all of the configured OCRs

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

Browsing My Oracle Support Knowledge Articles



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

Quiz

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.

Answer: a, b, d

Statements a, b, and d are correct.

Summary

In this lesson, you should have learned how to:

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

Practice 10: Overview

This practice covers working with:

- Log files
- `Ocrdump`
- `cluvfy`

11

Making Applications Highly Available with Oracle Clusterware

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

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.

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

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`**

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 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`.

ORACLE

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

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=/pathto/agent"
```


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`.

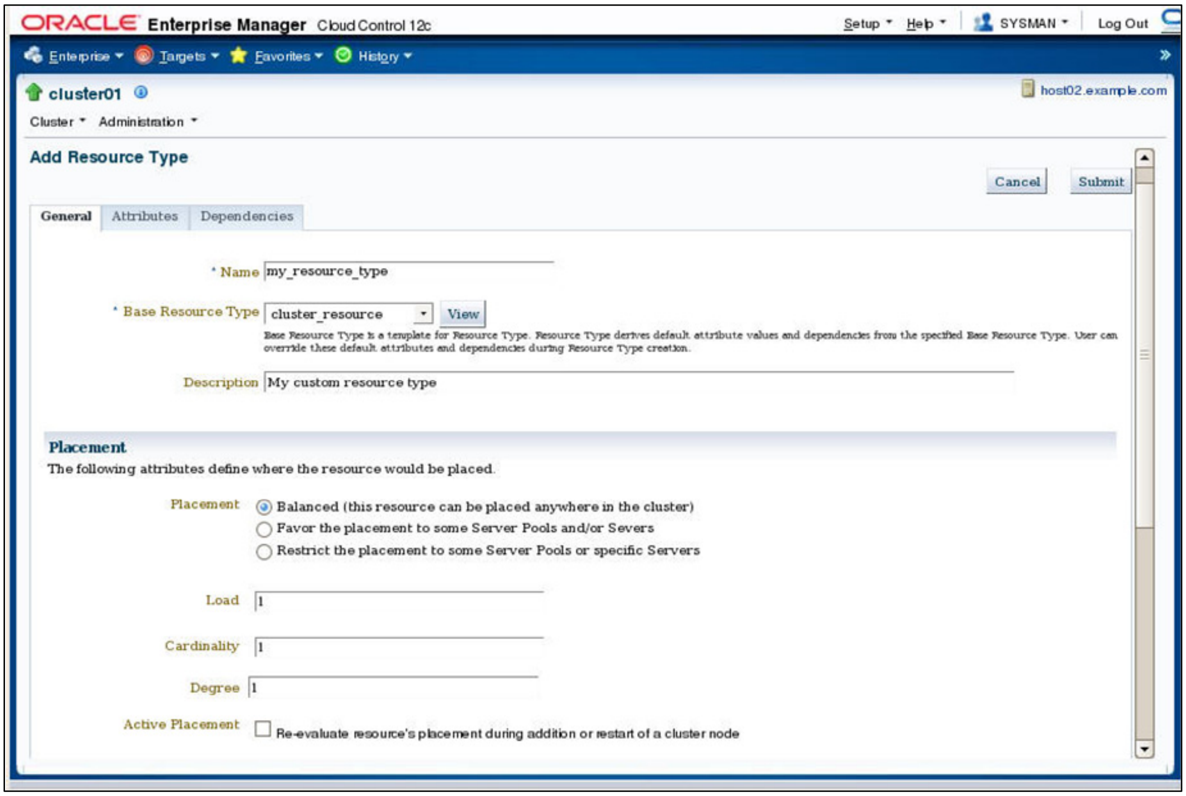
`-basetype` is the name of an existing base type. Any resource type that you create must either have `local_resource` or `cluster_resource` as its base resource type.

`-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



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 also. 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.

ORACLE

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 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
```


The resource attributes listed below 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	SCRIPT_TIMEOUT
ACTION_SCRIPT	ENABLED	SERVER_POOLS
ACTION_TIMEOUT	FAILURE_INTERVAL	SERVER_CATEGORY
ACTIONS	FAILURE_THRESHOLD	START_CONCURRENCY
ACTIVE_PLACEMENT	HOSTING_MEMBERS	START_DEPENDENCIES
AGENT_FILENAME	INSTANCE_FAILOVER	START_TIMEOUT
ALERT_TEMPLATE	INTERMEDIATE_TIMEOUT	STOP_CONCURRENCY
AUTO_START	LOAD	STOP_DEPENDENCIES
CARDINALITY	MODIFY_TIMEOUT	STOP_TIMEOUT
CHECK_INTERVAL	NAME	UPTIME_THRESHOLD
CHECK_TIMEOUT	OFFLINE_CHECK_INTERVAL	USER_WORKLOAD
CLEAN_TIMEOUT	RESTART_ATTEMPTS	USE_STICKINESS
DELETE_TIMEOUT	PLACEMENTRELOCATE_BY_DEPENDENCY	

The resource attributes listed below 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: Specify 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.

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.

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.

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 resource A and resource 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(intermedite: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`.

ORACLE

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(`**`intermedite`**`: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 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
...
```

ORACLE

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_nummber -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:

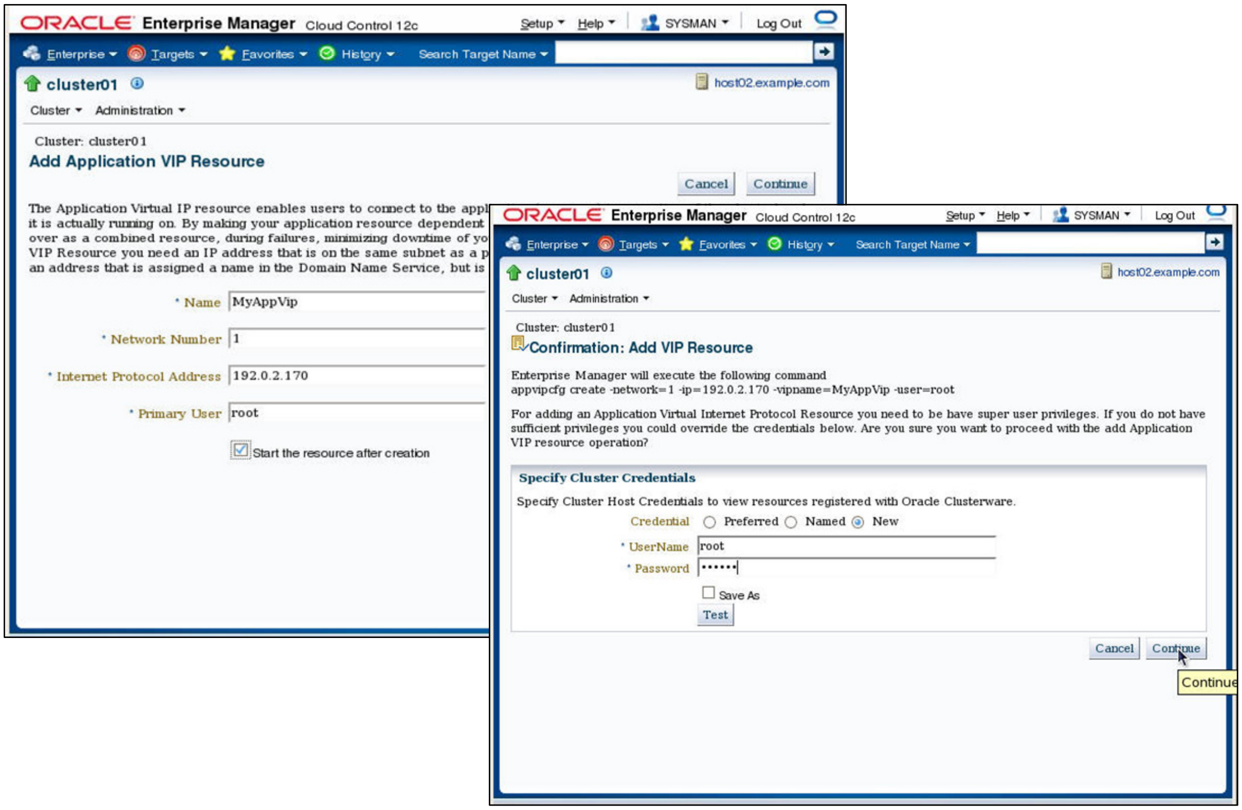
```
$ crsctl start resource appsVIP
```

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



ORACLE

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.

You must decide whether to use administrator or policy management for the application. Use administrator management for smaller, two-node 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. For example, if a resource only runs on node 1 and node 2 because only those nodes have the necessary files, then administrator management is probably more appropriate.

Oracle Clusterware supports the deployment of applications in access-controlled server pools made up of anonymous servers and strictly based on the desired pool size. Cluster policies defined by the administrator can and must be used in this case to govern the server assignment with desired sizes and levels of importance. Alternatively, a strict or preferred server assignment can be used, in which resources run on specifically named servers. This represents the pre-existing model available in earlier releases of Oracle Clusterware now known as administrator management.

Conceptually, a cluster hosting applications developed and deployed in both of the deployment schemes can be viewed as two logically separated groups of servers. One server group is used for server pools, enabling role separation and server capacity control. The other server group assumes a fixed assignment based on named servers in the cluster.

To manage an application using either deployment scheme, you must create a server pool before adding the resource to the cluster. A built-in server pool named Generic always owns the servers used by applications of administrator-based management. The Generic server pool is a logical division and can be used to separate the two parts of the cluster using different management schemes.

Registering a Resource

- Register resources in Oracle Clusterware 12*c* 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', ..."]
```


You can register resources in Oracle Clusterware 12*c* 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',"
```


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"
```

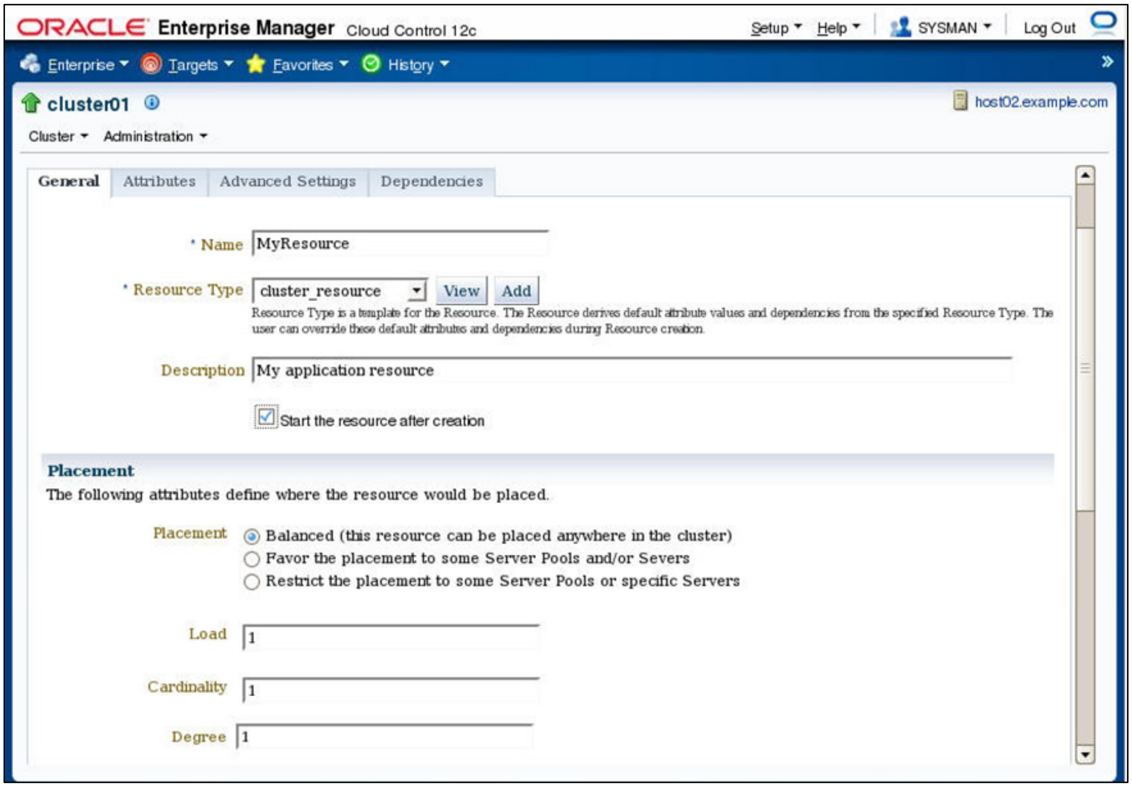
After you create the subpool, add the Apache web server resource, as follows:

```
$ 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)"
```

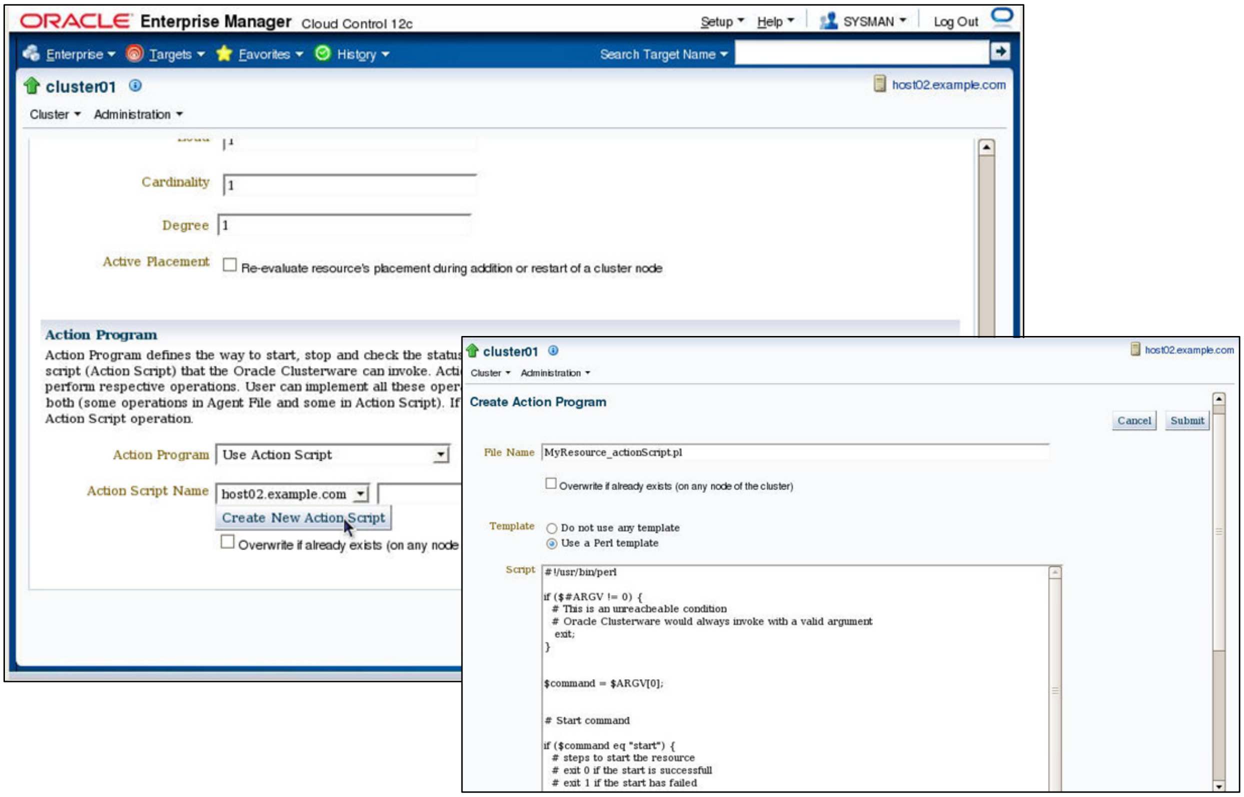
Adding Resources with EM



To add resources to Oracle Clusterware by using 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 Add Resource page.
5. Enter a name for the resource in the Name field.
6. Choose either `cluster_resource` or `local_resource` from the Resource Type drop-down list.
7. 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.
8. Select “Start the resource after creation” if you want the resource to start immediately.
9. The optional parameters in the Placement section define where in a cluster Oracle Clusterware places the resource.

Adding Resources with EM

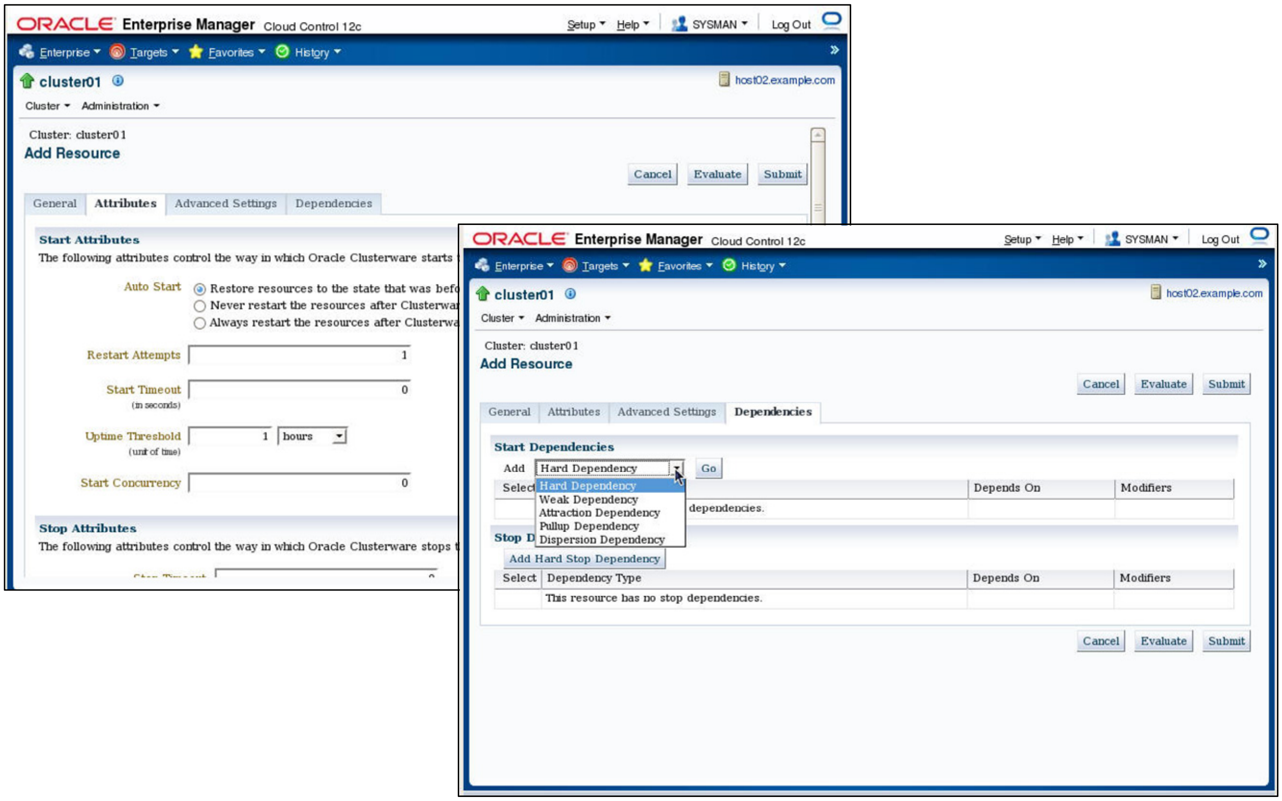


10. In the Action Program section, choose from the Action Program drop-down list whether Oracle Clusterware calls an action script, an agent file, or both to manage the resource.

    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



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

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 in tabular format:

```
# 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
```


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:

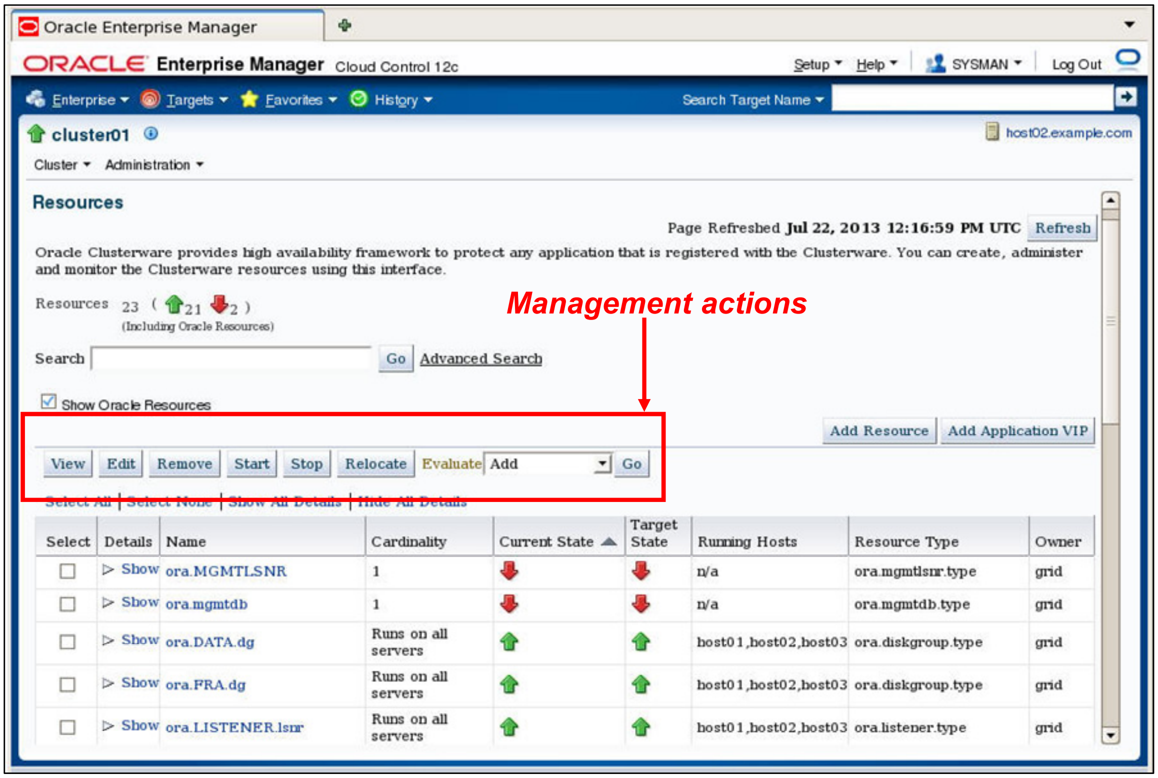
```
# crsctl status resource
```

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

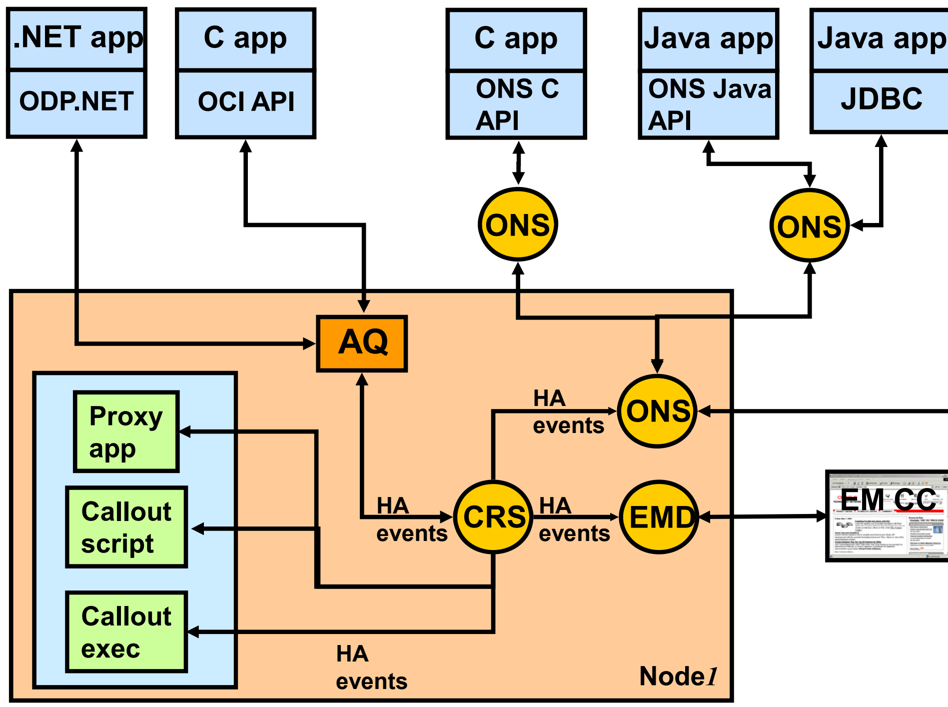


You can use Oracle Enterprise Manager to manage Oracle Clusterware resources. You can create and configure resources in Oracle Clusterware and also monitor and manage resources after they are deployed in the cluster. Resource Management tasks that can be performed from EM include:

- 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.

HA Events: ONS and FAN



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
```

- To add Oracle Notification Server:

```
$ srvctl add nodeapps -onslocalport 6100 -onsremoteport 6200
```

- To start or stop Oracle Notification Server:

```
$ srvctl start|stop nodeapps
```


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

Answer: a

The statement is true.

Quiz

Oracle Database 12*c* offers two methods for managing resources. These methods are:

a. Administration-based management
b. Threshold-based management
c. Policy-based management

Answer: a, c

Administration-based management and policy-based management are the correct answers.

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

Practice 11: Overview

This practice covers the following topics:

- Configuring highly available application resources on flex cluster leaf nodes
- Protecting the Apache application