Oracle® Database

2 Day + Real Application Clusters Guide

12c Release 2 (12.2)

E85152-02

August 2018

Oracle Database 2 Day + Real Application Clusters Guide, 12c Release 2 (12.2)

E85152-02

Primary Author: Janet Stern

Contributing Authors: Troy Anthony, Ian Cookson, Markus Michalewicz, Mark Scardina

Contributors: Eric Belden, Bernard Clouse, Carol Colrain, Jonathan Creighton, Rajesh Dasari, Vikash Gunreddy, Christopher Jones, Aparna Kamath, Aneesh Khandelwal, Nirmal Kumar, Venkat Maddali, Kuassi Mensah, Anil Nair, Hairong Qin, Kathy Rich, Alan Williams, Sivakumar Yarlagadda

Contents

Preface

Changes in This Release for Oracle Database 2 Day + Real Application Clusters Guide

1 Introduction to Deploying a Real Application Clusters Environment

2 Preparing Your Cluster

3 Installing Oracle Grid Infrastructure

4 Installing the Oracle Database Software and Creating a Database

5 Converting an Existing Database to Oracle RAC or Oracle RAC One Node

6 Administering Database Instances and Cluster Databases

7 Administering Oracle Clusterware Components

9 Administering Backup and Recovery

10 Monitoring and Troubleshooting Your Cluster

11 Adding and Deleting Nodes and Instances

12 Managing Oracle Software and Applying Patches

Glossary

List of Figures

List of Tables

Preface

Oracle Database 2 Day + Real Application Clusters Guide describes how to install, configure, and administer Oracle Clusterware, Oracle Automatic Storage Management (Oracle ASM), and Oracle Real Application Clusters (Oracle RAC) on a two-node system using the Oracle Linux operating system.

> **Note:**
>
> For Linux operating systems other than Oracle Linux, find the *Oracle Real Application Clusters Installation Guide* and *Oracle Grid Infrastructure Installation and Upgrade Guide* specific to your platform.

This guide covers topics that a reasonably knowledgeable Oracle database administrator (DBA) would need to know when moving from managing a single-instance Oracle Database environment to managing an Oracle RAC environment.

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

Oracle Database 2 Day + Real Application Clusters Guide is an Oracle RAC database administration guide for DBAs who want to install and use Oracle RAC. This guide assumes you have already read *Oracle Database 2 Day DBA*. This guide is intended for DBAs who:

- Want basic DBA skills for managing an Oracle RAC environment
- Manage Oracle databases for small- to medium-sized businesses

To use this guide, you should be familiar with the administrative procedures described in *Oracle Database 2 Day DBA*.

> **Note:**
>
> Some DBAs may be interested in moving the data from their single-instance Oracle Database to their Oracle RAC database. This guide also explains the procedures for doing this.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following in the Oracle Database documentation set:

- *Oracle Real Application Clusters Installation Guide for Linux and UNIX*
- *Oracle Grid Infrastructure Installation Guide for Linux*
- *Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Database 2 Day DBA*
- *Oracle Automatic Storage Management Administrator's Guide*

Conventions

The following text conventions are used in this guide:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Changes in This Release for Oracle Database 2 Day + Real Application Clusters Guide

This preface lists changes in *Oracle Database 2 Day + Real Application Clusters Guide*.

- Changes in Oracle Database 12c Release 2 (12.2)
- Changes in Oracle Database 12c Release 1 (12.1)

Changes in Oracle Database 12*c* Release 2 (12.2)

The following are changes in *Oracle Database 2 Day + Real Application Clusters Guide* for Oracle Database 12*c* Release 2 (12.2).

- New Features for Oracle Database 12c Release 2 (12.2)
- Deprecated Features
- Desupported Features for Oracle Database 12c Release 2 (12.2)

New Features for Oracle Database 12*c* Release 2 (12.2)

The following features are new in this release:

- Application Continuity for Planned Database Maintenance

  This feature hides scheduled maintenance operations that are required for the underlying infrastructure (Oracle Database, Oracle Grid Infrastructure, operating system and hardware) without placing a burden on application developers. This feature enables you to conduct scheduled maintenance without interruption to most application work.

  See Managing Planned Maintenance Without User Interruption.
- Application Continuity for OCI Applications

  Application Continuity is an application-independent feature that attempts to recover incomplete requests from an application perspective and masks many system, communication and hardware failures, and storage outages from the user. Application Continuity is supported when used with OCI Session pools, Tuxedo, the JDBC Replay data source or OCI driver (when you add request boundaries with your own connection pool) and OCI-based applications such as SQL*Plus.

  See Preparing to Use Application Continuity.
- Application Continuity Support for ODP .NET Unmanaged Driver

Application Continuity recovers incomplete requests from an ODP.NET Unmanaged Driver perspective and masks many system failures, communication failures, hardware failures, and storage outages from the user. Application Continuity ensures that transactions are executed no more than once. When failures do occur, they are generally masked from the user. This feature leads to improved user experience, higher application availability, and improved ODP.NET developer productivity.

See Preparing to Use Application Continuity.

- Automatic Configuration of Oracle ASM Filter Driver (Oracle ASMFD)

  Oracle ASMFD simplifies the configuration and management of disk devices by eliminating the need to rebind disk devices used with Oracle ASM each time the system is restarted. The configuration for Oracle ASMFD can now be enabled with a check box to be an automated process during Oracle Grid Infrastructure installation.

  See Installing Oracle Grid Infrastructure for a New Cluster.

- Cluster Health Advisor

  The Cluster Health Advisor provides system administrators and database administrators early warning of pending performance issues and root causes and corrective actions for Oracle RAC databases and cluster nodes. This advanced proactive diagnostic capability increases availability and performance management.

  See Cluster Health Advisor.

- Java APIs for FAN Events (UP, DOWN and Load Balancing Advisory Events)

  The Oracle Fast Application Notification (FAN) library (simplefan.jar) is enhanced with the capability to identify UP events. The `oracle.simplefan.FanUpEventListener` interface is enhanced with two methods, `NodeUpEvent()` and `ServiceUpEvent()`. `ServiceDownEvent`, `LoadAdvisoryEvent`, and `NodeDownEvent` were added in a previous release. Java containers, frameworks, and applications looking to handle FAN events can use these APIs to subscribe to Oracle Database RAC FAN events for building high availability solutions.

  See *Oracle Database RAC FAN Events Java API Reference*.

- JDBC Driver Support for Fast Application Notification (FAN)

  The Java Database Connectivity (JDBC) driver now supports Oracle Database RAC FAN events for enhanced support for planned maintenance and unplanned down times:

  – `oracle.jdbc.fanEnabled`: A system property to enable or disable the FAN support in the driver. If Oracle Universal Connection Pool (UCP) is used as a client-side pool, then UCP takes the precedence.

  – `oracle.jdbc.fanONSConfig`: A connection property to be used by the driver as remote Oracle Notification Services (ONS) configuration. This property is required only for the pre-12c database versions.

  See Configuring Application Clients for High Availability.

- Provisioning of Oracle Grid Infrastructure Homes

  Oracle Database 12*c* release 2 (12.2) supports provisioning, patching, and upgrade of Grid Infrastructure homes using Rapid Home Provisioning.

See Rapid Home Provisioning, Patching, and Upgrading.

- Separation of Duty for Administering Oracle Real Application Clusters

  Starting with Oracle Database 12*c* release 2 (12.2), Oracle Database provides support for separation of duty best practices when administering Oracle Real Application Clusters (Oracle RAC) by introducing the SYSRAC administrative privilege for the clusterware agent. This feature removes the need to use the powerful SYSDBA administrative privilege for Oracle RAC.

  SYSRAC, like SYSDG, SYSBACKUP and SYSKM, helps enforce separation of duties and reduce reliance on the use of SYSDBA on production systems. This administrative privilege is the default mode for connecting to the database by the clusterware agent on behalf of the Oracle RAC utilities such as `srvctl`.

  See About Operating System Users and Groups.

- Support for IPv6 Based IP Addresses for the Oracle Cluster Interconnect

  Starting with Oracle Grid Infrastructure 12*c* release 2 (12.2), you can use either IPv4 or IPv6 based IP addresses to configure cluster nodes on the private network. You can use more than one private network for the cluster.

- Zip Image based Grid Infrastructure Installation

  Starting with Oracle Grid Infrastructure 12*c* release 2 (12.2), the installation media is replaced with a zip file for the Oracle Grid Infrastructure installer. Run the installation wizard after extracting the zip file into the target home path.

  See Installing Oracle Grid Infrastructure for a New Cluster.

Deprecated Features

The following features are deprecated in this release, and may be desupported in another release. See *Oracle Database Upgrade Guide* for a complete list of deprecated features in this release.

- Deprecation of Direct File System Placement for Oracle Cluster Registry (OCR) and Voting Files

  Starting with Oracle Grid Infrastructure 12*c* Release 2 (12.2), the placement of Oracle Clusterware files: the Oracle Cluster Registry (OCR), and the Voting Files, directly on a shared file system is deprecated in favor of having Oracle Clusterware files managed by Oracle Automatic Storage Management (Oracle ASM). Placing Oracle Clusterware files directly on a shared file system may be subject to desupport in future releases. If you need to use a supported shared file system, either a Network File System, or a shared cluster file system instead of native disks devices, then you must create Oracle ASM disks on the shared file systems that you plan to use for hosting Oracle Clusterware files before installing Oracle Grid Infrastructure. You can then use the Oracle ASM disks in an Oracle ASM disk group to manage Oracle Clusterware files.

- Deprecation of `configToolAllCommands` script

  The `configToolAllCommands` script runs in the response file mode to configure Oracle products after installation and uses a separate password response file. Starting with Oracle Database 12*c* Release 2 (12.2), the `configToolAllCommands` script is deprecated and is subject to desupport in a future release.

To perform postinstallation configuration of Oracle products, you can now run the Oracle Database or Oracle Grid Infrastructure installer with the -`executeConfigTools` option. You can use the same response file created during installation to complete postinstallation configuration.

See *Oracle Real Application Clusters Installation Guide for Linux and UNIX*.

Desupported Features for Oracle Database 12*c* Release 2 (12.2)

Some features previously described in this document are desupported in Oracle Database 12*c* Release 2 (12.2). See *Oracle Database Upgrade Guide* for a list of desupported features.

Changes in Oracle Database 12*c* Release 1 (12.1)

The following are changes in *Oracle Database 2 Day + Real Application Clusters Guide* for Oracle Database 12*c* Release 1 (12.1).

- New Features
- Desupported Features

New Features

The following features are new in this release:

- Oracle Enterprise Manager Database Express replaces Oracle Enterprise Manager Database Control (DB Control)

  In previous Oracle Database releases, DB Control was the primary database management tool described in this manual. DB Control is not included in this release.

  See *Oracle Database 2 Day DBA*.

- Oracle Home User Support for Oracle Database

  Starting with Oracle Database 12*c* release 1 (12.1), Oracle Database supports the use of an Oracle Home User, which can be specified at installation time. The Oracle Home User can be a Built-in Account or a Windows User Account. If you specify a Windows User Account, then the user should be a low privileged (non-Administrator) account to ensure that the Oracle Home User has a limited set of privileges. Using an Oracle Home User ensures that Oracle Database services have only those privileges required to run Oracle products.

  See *Oracle Database Platform Guide for Microsoft Windows*.

- Application Continuity

  Application Continuity attempts to mask outages from end users and applications by recovering requests following recoverable outages, unplanned and planned. Application Continuity performs this recovery beneath the application so that the outage appears to the application as a delayed execution. Application Continuity masks recoverable outages - those in which requests would succeed if they were reissued. Examples include system failures, network disconnects, foreground failures, and storage failures.

  See *Oracle Real Application Clusters Administration and Deployment Guide*.

- Transaction Guard

  Before Oracle Database 12*c* release 1 (12.1), it was difficult for a database application to recover after an outage, because the commit message that Oracle Database (the server) returned to the application (the client) was not persistent. If the connection between Oracle Database and the application broke or if the database session became unavailable, then the application received a disconnection error message. After an outage, if an application user resubmitted an in-flight transaction that had been committed, then duplicate transactions resulted.

  The Oracle Database feature Transaction Guard ensures that each transaction executes at most once. Its PL/SQL interface, the `DBMS_APP_CONT.GET_LTXID_OUTCOME` procedure, enables an application to determine the outcome of the in-flight transaction after an outage and then recover any work that was lost due to the outage.

  See *Oracle Database Development Guide*.

- Java and JDBC Support for Application Continuity

  Application Continuity for Java is available with Oracle Database, JDBC Thin driver, and the Oracle connection pools: UCP (Universal Connection Pool) and WebLogic Server Active GridLink.

  Application Continuity is transparent for Java EE and Java SE applications that use Oracle JDBC, use Oracle connection pools (UCP or WLS Active GridLink), and do not have external actions. For applications with external actions (for example, using autonomous transactions or using UTL_HTTP to issue an SOA call), Application Continuity is transparent only if the application's correctness is preserved when these external actions are replayed after a failure.

  See *Oracle Database JDBC Developer’s Guide*.

- Oracle JDBC Support for Transaction Guard

  See *Oracle Database JDBC Developer’s Guide*.

Desupported Features

Some features previously described in this document are desupported in Oracle Database 12*c* Release 1 (12.1). See *Oracle Database Upgrade Guide* for a list of desupported features.

1 Introduction to Deploying a Real Application Clusters Environment

Oracle Real Application Clusters (Oracle RAC) enables an Oracle database to run across a cluster of servers, providing fault tolerance, performance, and scalability with no application changes necessary.

Oracle RAC provides high availability for applications by removing the single point of failure with a single server.

1.1 About the 2 Day + Real Application Clusters Guide

This guide is a task-oriented guide for Oracle Real Application Clusters (Oracle RAC) database administration.

This guide shows you how to configure and manage the environment for Oracle Clusterware and Oracle RAC. The administration procedures and examples in this guide assume Oracle Linux is the operating system. For the purposes of this guide, a two-node cluster is used in the procedures and examples.

The goal of this guide is to help you understand the basic steps required to install and maintain an Oracle RAC environment, including how to perform basic troubleshooting, performance monitoring, and backup and recovery activities. This guide is based on Oracle Linux, but you do not have to be a Linux expert to use this guide.

- What This Guide Is Not
  The *2 Day + Real Application Clusters Guide* is not a comprehensive description of Oracle Real Application Clusters (Oracle RAC). It describes concepts only when necessary for completing a particular task.
- Related Materials
  The *2 Day + Real Application Clusters Guide* is part of a comprehensive set of learning materials for administering Oracle Databases
- Overview of Oracle RAC Documentation
  The *2 Day + Real Application Clusters Guide* describes how to install, configure, and manage Oracle Real Application Clusters (Oracle RAC) and Oracle Clusterware.

1.1.1 What This Guide Is Not

The *2 Day + Real Application Clusters Guide* is not a comprehensive description of Oracle Real Application Clusters (Oracle RAC). It describes concepts only when necessary for completing a particular task.

> **See Also:**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for a detailed description of Oracle RAC concepts

1.1.2 Related Materials

The *2 Day + Real Application Clusters Guide* is part of a comprehensive set of learning materials for administering Oracle Databases

The learning materials include a 2 Day DBA Oracle By Example (OBE) series, available on Oracle Technology Network, and Oracle University instructor-led classes. The OBE series also has animated demos you view using a web browser.

You can view the OBE content for Oracle Database and Oracle RAC at the following website:

`http://www.oracle.com/technetwork/tutorials/`

1.1.3 Overview of Oracle RAC Documentation

The *2 Day + Real Application Clusters Guide* describes how to install, configure, and manage Oracle Real Application Clusters (Oracle RAC) and Oracle Clusterware.

This guide provides examples for how you could deploy Oracle RAC on a two-node cluster, using the Oracle Linux operating system. Oracle Linux is fully compatible — both source and binary — with Red Hat Enterprise Linux. It includes the identical set of packages at the same version levels with the same source code as the Red Hat distribution.

This guide is for DBAs who have experience with single-instance Oracle environments and have read *Oracle Database 2 Day DBA*.

- Useful Oracle RAC Guides
  Oracle provides a comprehensive set of documentation for Oracle Real Application Clusters (Oracle RAC).
- Other Useful Oracle Documentation
  In addition to the *Oracle Database 2 Day DBA* guide, there are many other books that are referenced when performing DBA tasks.

1.1.3.1 Useful Oracle RAC Guides

Oracle provides a comprehensive set of documentation for Oracle Real Application Clusters (Oracle RAC).

The following list contains useful related documentation:

- *Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Clusterware Administration and Deployment Guide*
- *Oracle Real Application Clusters Installation Guide for Linux and UNIX* (or other operating system)
- *Oracle Grid Infrastructure Installation and Upgrade Guide for Linux* (or other operating system)
- *Oracle Autonomous Health Framework User's Guide*

1.1.3.2 Other Useful Oracle Documentation

In addition to the *Oracle Database 2 Day DBA* guide, there are many other books that are referenced when performing DBA tasks.

The following list displays other books in the Oracle documentation library that were referenced within this guide:

- *Oracle Automatic Storage Management Administrator's Guide*
- *Oracle Database Concepts*
- *Oracle Database Administrator’s Guide*
- *Oracle Database Net Services Administrator's Guide*
- *Oracle Database Administrator's Reference for Linux and UNIX-Based Operating Systems*
- *Oracle Database Licensing Information*
- *Oracle Database Reference*
- *Oracle Database Backup and Recovery User’s Guide*
- *Oracle Database 2 Day + Performance Tuning Guide*
- *Oracle Database Performance Tuning Guide*
- *Oracle Universal Installer (OUI) User’s Guide*
- *Oracle Database Upgrade Guide*

1.2 About Oracle Grid Infrastructure for a Cluster and Oracle RAC

Oracle Grid Infrastructure for a cluster refers to the installation of the combined products. Oracle Clusterware and Oracle ASM are still individual products, and are referred to by those names.

Oracle Clusterware and Oracle ASM are installed into a single home directory, which is called the Grid home.

Oracle Clusterware enables servers, referred to as **hosts** or nodes, to operate as if they are one server, commonly referred to as a cluster. Although the servers are standalone servers, each server has additional processes that communicate with other servers. In this way the separate servers appear as if they are one server to applications and end users. Oracle Clusterware provides the infrastructure necessary to run Oracle RAC. The combined processing power of the multiple servers provides greater availability, throughput, and scalability than is available from a single server.

The nodes in a cluster can be organized into a server pool for better resource management. Each server pool has the following properties:

- The minimum number of nodes that should be in the server pool
- The maximum number of nodes that can be in the server pool
- The relative importance of this server pool to other server pools

Upon installation of Oracle Grid Infrastructure for a cluster, a default server pool, called the Free pool, is created automatically. All servers in a new installation are assigned to the Free server pool, initially. If you create a new server pool, then the servers move from the Free pool to the new server pool automatically.

When you create an Oracle RAC database that is a policy-managed database, you specify the number of servers that are needed for the database, and a server pool is automatically created for the database. Oracle Clusterware populates the server pool with the servers it has available. If you do not use server pools, then you create an administrator-managed database.

> **See Also:**
>
> - *Oracle Real Application Clusters Administration and Deployment Guide*
> - *Oracle Clusterware Administration and Deployment Guide*

1.3 About Oracle Automatic Storage Management

Oracle Automatic Storage Management (Oracle ASM) is an integrated, high-performance volume manager and file system.

Oracle ASM supports storing the Oracle Clusterware OCR and voting disk files, and also provides a general purpose cluster file system called Oracle Automatic Storage Management Cluster File System (Oracle ACFS). Oracle ASM is based on the principle that the database should manage storage instead of requiring an administrator to do it. Oracle ASM eliminates the need for you to directly manage potentially thousands of Oracle database files.

- About Using Oracle ASM Disk Groups with Oracle RAC
  Oracle ASM groups the disks in your storage system into one or more disk groups. You manage a small set of disk groups and Oracle ASM automates the placement of the database files within those disk groups.
- Benefits of Using Oracle ASM
  Oracle ASM reduces the administrative overhead for managing database storage by consolidating data storage into a small number of disk groups. Oracle ASM disk groups provide a number of additional benefits.
- About Oracle Flex ASM
  Oracle Flex ASM enables an Oracle ASM instance to run on a separate physical server from the database servers.
- About Oracle ACFS
  Oracle Automatic Storage Management Cluster File System (Oracle ACFS) is a multi-platform, scalable file system, and storage management technology that extends Oracle Automatic Storage Management (Oracle ASM) functionality to support all customer files.
- About Oracle ASM Running in an Oracle RAC Environment
  In an Oracle RAC environment, Oracle ASM has multiple instances which run on one or more nodes in the cluster.

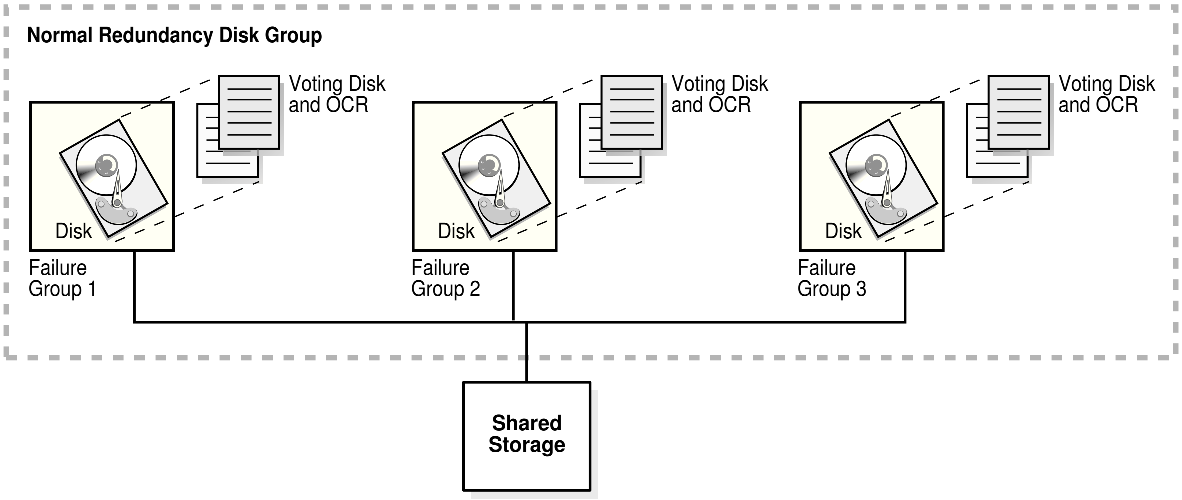
1.3.1 About Using Oracle ASM Disk Groups with Oracle RAC

Oracle ASM groups the disks in your storage system into one or more disk groups. You manage a small set of disk groups and Oracle ASM automates the placement of the database files within those disk groups.

Oracle Cluster Registry (OCR) and voting disks must be placed on Oracle ASM disk groups. When using Oracle Real Application Clusters (Oracle RAC), each instance must have access to the data files and recovery files for the Oracle RAC database. Using Oracle ASM is an easy way to satisfy this requirement.

In the following figure, the normal redundancy disk group stored in Oracle ASM contains three failure groups (shown as three disks) to store the voting disk and OCR files for Oracle Clusterware. Because the voting disk records which nodes are currently members of the cluster, there must be at least three voting disks for a two-node cluster. If the Oracle ASM disk group did not contain the voting disk and OCR files, then you could have a normal redundancy disk group with only two failure groups.

Figure 1-1 Oracle Clusterware Files Stored in an Oracle ASM Disk Group



Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*
- About Oracle Clusterware
  Oracle Clusterware includes a high availability framework for managing any application that runs on your cluster. Oracle Real Application Clusters (Oracle RAC) uses Oracle Clusterware as the infrastructure that binds multiple nodes that then operate as a single server.

1.3.2 Benefits of Using Oracle ASM

Oracle ASM reduces the administrative overhead for managing database storage by consolidating data storage into a small number of disk groups. Oracle ASM disk groups provide a number of additional benefits.

Oracle ASM provides the following benefits:

- **Striping**—Oracle ASM spreads data evenly across all disks in a disk group to optimize performance and utilization. This even distribution of database files eliminates the need for regular monitoring and I/O performance tuning.
- **Mirroring**—Oracle ASM increases data availability by optionally mirroring any file. Oracle ASM mirrors at the file level, unlike operating system mirroring, which mirrors at the disk level. Mirroring means keeping redundant copies, or mirrored copies, of each extent of the file, to help avoid data loss caused by disk failures. The mirrored copy of each file extent is always kept on a different disk from the original copy. If a disk fails, then Oracle ASM continues to access affected files by accessing mirrored copies on the surviving disks in the disk group.
- **Online storage reconfiguration and dynamic rebalancing**—When using Oracle ASM, you can add or remove disks from your disk storage system while the database is operating. When you add a disk to a disk group, Oracle ASM automatically redistributes the data so that it is evenly spread across all disks in the disk group, including the new disk. The process of redistributing data so that it is also spread across the newly added disks is known as **rebalancing**. It is done in the background and with minimal impact to database performance.

- **Managed file creation and deletion**—Oracle ASM further reduces administration tasks by enabling files stored in Oracle ASM disk groups to be managed by Oracle Database. Oracle ASM automatically assigns file names when files are created, and automatically deletes files when they are no longer needed by the database.

1.3.3 About Oracle Flex ASM

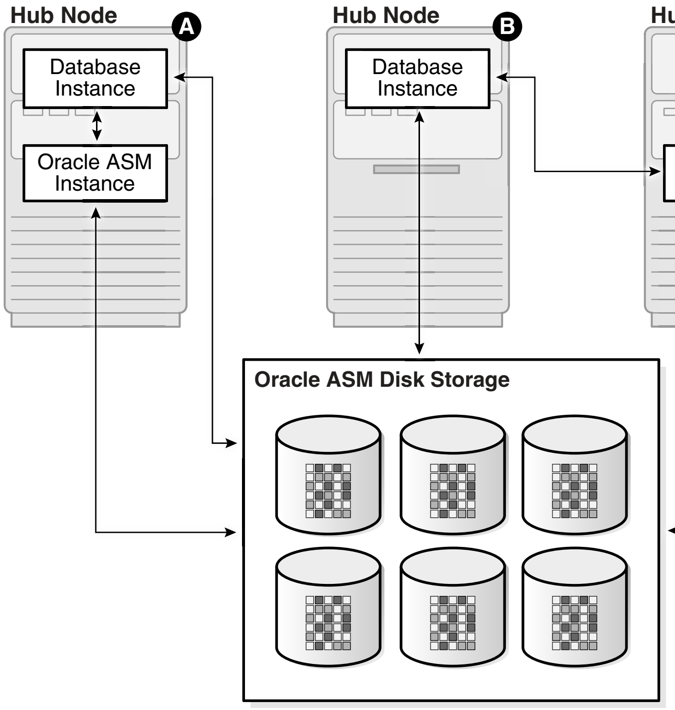
Oracle Flex ASM enables an Oracle ASM instance to run on a separate physical server from the database servers.

With Oracle Flex ASM, you can consolidate all the storage requirements into a single set of disk groups. All these disk groups are mounted by and managed by a small set of Oracle ASM instances running in a single cluster. With this deployment, larger clusters of Oracle ASM instances can support more database clients while reducing the Oracle ASM footprint for the overall system. You can specify the number of Oracle ASM instances with a cardinality setting. The default is three instances.

For the Oracle RAC deployment described in this guide, Oracle Flex ASM is configured as local Oracle ASM clients with direct access to Oracle ASM disk. This supports the previous standard architecture in which database clients run with an Oracle ASM instance on the same server. Each server must be configured as a Hub node in the cluster. Communication with the Oracle ASM instances is performed using a private network interface, which can be either a dedicated Oracle ASM network interface, or the same network interface as the cluster private network.

In a larger cluster with Oracle Flex ASM you can have nodes running database instances, but not Oracle ASM instances. Database clients that are running on Hub nodes of the Oracle ASM cluster access Oracle ASM remotely for metadata, but perform block I/O operations directly to Oracle ASM disks. A Hub node is a node in an Oracle ASM cluster that is tightly connected with other servers and has direct access to a shared disk. Depending on the distribution of database instances and Oracle ASM instances, a database client may access Oracle ASM locally on the same node or remotely over the Oracle ASM network.

Figure 1-2   Oracle ASM Flex Client Configurations



1.3.4 About Oracle ACFS

Oracle Automatic Storage Management Cluster File System (Oracle ACFS) is a multi-platform, scalable file system, and storage management technology that extends Oracle Automatic Storage Management (Oracle ASM) functionality to support all customer files.

Oracle ACFS supports Oracle Database files and application files, including executables, database data files, database trace files, database alert logs, application reports, BFILEs, and configuration files. Other supported files are video, audio, text, images, engineering drawings, and all other general-purpose application file data. Oracle ACFS conforms to POSIX standards for Linux and UNIX, and to Windows standards for Windows.

An Oracle ACFS file system communicates with Oracle ASM and is configured with Oracle ASM storage. Oracle ACFS leverages Oracle ASM functionality that enables:

- Oracle ACFS dynamic file system resizing
- Maximized performance through direct access to Oracle ASM disk group storage
- Balanced distribution of Oracle ACFS across Oracle ASM disk group storage for increased I/O parallelism
- Data reliability through Oracle ASM mirroring protection mechanisms

1.3.5 About Oracle ASM Running in an Oracle RAC Environment

In an Oracle RAC environment, Oracle ASM has multiple instances which run on one or more nodes in the cluster.

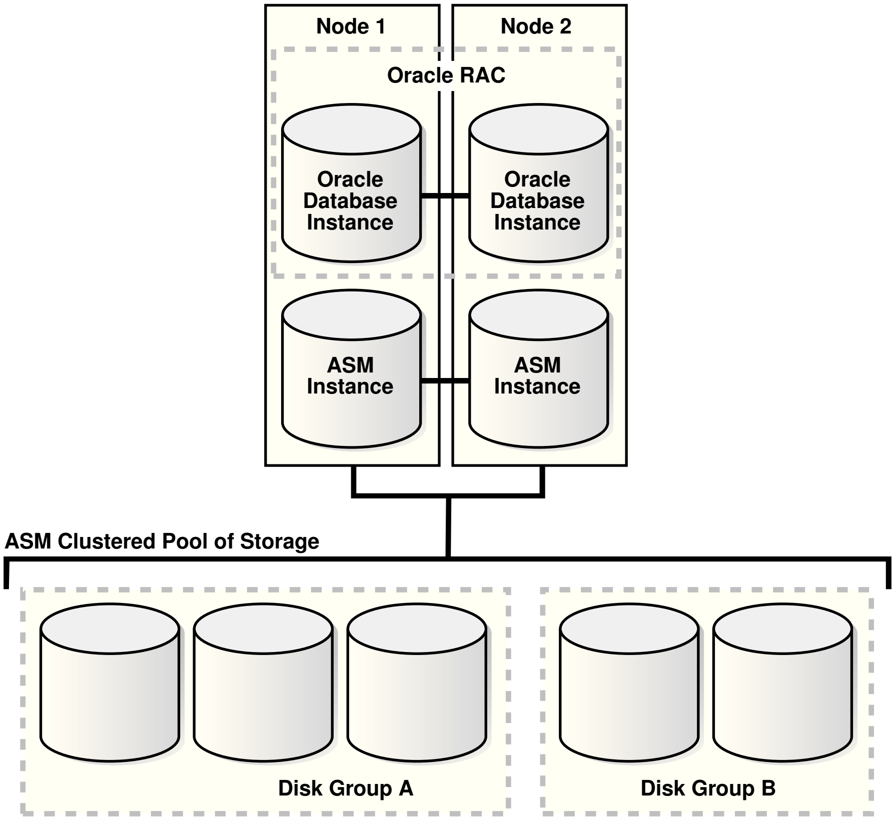
Oracle ASM is implemented as a special kind of Oracle instance, with its own System Global Area and background processes.

Starting with Oracle Grid Infrastructure 12*c* Release 1 (12.1), as part of an Oracle Flex Cluster installation, Oracle ASM is installed in an Oracle Flex ASM cluster. Each Oracle Flex ASM cluster has its own name that is globally unique within the enterprise. Every Oracle Flex ASM cluster has one or more Hub Nodes on which Oracle ASM instances are running

Oracle Flex ASM enables an Oracle ASM instance to run on a separate physical server from the database servers. Many Oracle ASM instances can be clustered to support numerous database clients. You can consolidate all the storage requirements into a single set of disk groups. All these disk groups are managed by a small set of Oracle ASM instances running in a single Oracle Flex Cluster.

The following figure shows the storage configuration commonly used for a 2-node Oracle RAC database.

Figure 1-3   Oracle Instances and Oracle ASM Storage



Oracle recommends that you use Oracle ASM for your Oracle Clusterware files and Oracle RAC data files. Oracle databases can use both Oracle ASM files and non-Oracle ASM files. You can also create a file system using Oracle ACFS to store your database Oracle Home and any other external (non-database) files in the cluster.

See Also:

- About Oracle Clusterware for information about Oracle Clusterware files
- *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM concepts and features

1.4 About Oracle Real Application Clusters

Oracle RAC extends Oracle Database so that you can store, update, and efficiently retrieve data using multiple database instances on different servers at the same time.

Oracle RAC provides the software that manages multiple servers and instances as a single group. The data files that comprise the database must reside on shared storage that is accessible from all servers that are part of the cluster. Each server in the cluster runs the Oracle RAC software.

An Oracle database has a one-to-one relationship between data files and the database instance. An Oracle RAC database, however, has a one-to-many relationship between data files and database instances. In an Oracle RAC database, multiple instances access a single set of database files.

Each database instance in an Oracle RAC database uses its own memory structures and background processes. Oracle RAC uses Cache Fusion to synchronize the data stored in the buffer cache of each database instance. Cache Fusion moves current data blocks (which reside in memory) between database instances, rather than having one database instance write the data blocks to disk and requiring another database instance to reread the data blocks from disk. When one instance requires a data block located in the buffer cache of another instance, Cache Fusion transfers the data block directly between the instances using the interconnect , enabling the Oracle RAC database to access and modify data as if the data resided in a single buffer cache.

Oracle RAC is also a key component for implementing the enterprise grid computing architecture using Oracle software. Having multiple database instances accessing a single set of data files prevents the server from being a single point of failure. If a node in the cluster fails, then the Oracle database continues running on the remaining nodes. Individual nodes can be stopped for maintenance while application users continue to work.

Oracle RAC supports mainstream business applications, such as OLTP, DSS, and also popular packaged products such as SAP, PeopleSoft, Siebel, and Oracle E-Business Suite, and custom applications. Any packaged or custom application that scales on an Oracle database scales well on Oracle RAC without requiring changes to the application code.

You will learn more about the operation of the Oracle RAC database in a cluster, how to build the cluster, and the structure of an Oracle RAC database in other sections of this guide.

- About Oracle RAC One Node
  Oracle RAC One Node is a single instance of an Oracle RAC-enabled database running on one node in the cluster, only, under normal operations.

- About Oracle Multitenant with Oracle RAC
  Oracle Multitenant is based on an architecture that allows a multitenant container database (CDB) to hold several pluggable databases (PDBs).
- About Server Pools
  Understand the server pool concepts in Oracle Real Application Clusters (Oracle RAC) environments.
- About Installing Oracle RAC on Different Operating Systems
  Oracle Real Application Clusters is supported on a variety of different operating systems.

> **See Also:**
>
> - *Oracle Real Application Clusters Administration and Deployment Guide* for more information about Oracle RAC concepts and architecture
> - *Oracle Clusterware Administration and Deployment Guide* for more information about Oracle Clusterware concepts and architecture

1.4.1 About Oracle RAC One Node

Oracle RAC One Node is a single instance of an Oracle RAC-enabled database running on one node in the cluster, only, under normal operations.

Oracle Real Application Clusters One Node (Oracle RAC One Node) is an option to Oracle Database Enterprise Edition available since Oracle Database 11*g* release 2 (11.2). This option adds to the flexibility that Oracle offers for database consolidation while reducing management overhead by providing a standard deployment for Oracle databases in the enterprise. Oracle RAC One Node database requires Oracle Grid Infrastructure and, therefore, requires the same hardware setup as an Oracle RAC database.

Unlike Oracle RAC, Oracle RAC One Node normally has only one instance that accesses a shared set of data files, also known as the database. This database is fully Oracle RAC-enabled, but does not span multiple hardware systems at the same time. Instead, the Oracle RAC One Node database instance fails over to another server in the cluster if a server, instance or a related and monitored component on this server fails. With Oracle RAC One Node, there is no limit to server scalability and, if applications grow to require more resources than a single node can supply, then you can upgrade your applications online to Oracle RAC. If the node that is running Oracle RAC One Node becomes overloaded, then you can relocate the instance to another node in the cluster.

For cases of planned downtime such as operating system or database patching, Oracle RAC One Node provides a unique feature, Online Database Relocation, which allows relocating a database from one server to another without interrupting the database service. With Oracle RAC One Node you can use the Online Database Relocation feature to relocate the database instance with no downtime for application users.

See Also:

- "About Oracle Real Application Clusters"
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about Oracle RAC One Node

1.4.2 About Oracle Multitenant with Oracle RAC

Oracle Multitenant is based on an architecture that allows a multitenant container database (CDB) to hold several pluggable databases (PDBs).

Oracle Multitenant is an option with Oracle Database 12*c* that simplifies consolidation, provisioning, upgrades, and more. You can adopt an existing database as a PDB without having to change the application tier. In this architecture, Oracle RAC provides the local high availability that is required when consolidating various business-critical applications on one system.

When using PDBs with Oracle RAC, the multitenant CDB is based on Oracle RAC. This means for the CDB database, there is an instance running on multiple nodes in the cluster. You can make each PDB available on either every instance of the Oracle RAC CDB or a subset of instances. In either case, access to and management of the PDBs are regulated using dynamic database services, which are also used by applications to connect to the respective PDB.

If you create an Oracle RAC database as a CDB and plug one or more PDBs into the CDB, then, by default, a PDB is not started automatically on any instance of the Oracle RAC CDB. When you assign the first dynamic database service to the PDB (other than the default database service which has the same name as the database name), then the PDB is made available on those instances on which the service runs.

Whether a PDB is available on more than one instance of an Oracle RAC CDB, the CDB is typically managed by the services running on the PDB. You can manually enable PDB access on each instance of an Oracle RAC CDB by starting the PDB manually on that instance.

1.4.3 About Server Pools

Understand the server pool concepts in Oracle Real Application Clusters (Oracle RAC) environments.

When Oracle Clusterware is installed, two internal 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 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.

Oracle RAC databases support two types of server pool management styles and deployment models: Policy-based management and administrator-managed.

If you create an Oracle RAC One Node database, note the following:

- Oracle RAC One Node runs only in one server pool. This server pool is treated the same as any other server pool.
- Online relocation of an Oracle RAC One Node database instance permits planned migrations of an Oracle RAC One Node database from one node to another node. Relocations must always be within a server pool.

Policy-Based Management of Oracle RAC Databases

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 non-database use, such as creating a server pool for servers hosting an application.

You can configure each server pool with a minimum and maximum size, which determines scalability. Oracle Clusterware manages availability between server pools, and you can further regulate availability by configuring the importance value of individual server pools.

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

Administrator-Managed Oracle RAC Databases

Deployment is based on the Oracle RAC deployment types that existed before Oracle Database 11*g* release 2 (11.2) and requires that you statically configure each database instance to run on a specific node in the cluster, and that you configure database services to run on specific instances belonging to a certain database using the preferred and available designation.

When you review the database resource for an administrator-managed database, you see a server pool defined with the same name as the Oracle database. This server pool is part of a special Oracle-defined server pool called Generic. Oracle RAC manages the Generic server pool to support administrator-managed databases. When you add or remove an administrator-managed database using either SRVCTL or DBCA, Oracle RAC creates or removes the server pools that are members of Generic.

Related Topics

- *Oracle Clusterware Administration and Deployment Guide*

1.4.4 About Installing Oracle RAC on Different Operating Systems

Oracle Real Application Clusters is supported on a variety of different operating systems.

If you plan to install and configure Oracle RAC on an operating system other than Oracle Linux, then you can still use this guide to obtain a general understanding about how to deploy Oracle RAC. You can also use this guide for deploying Oracle RAC on clusters with more than two nodes. For all environments that do not match the environment that this guide describes, modify the examples accordingly.

When installing Oracle RAC on a different platform or different operating system version than Oracle Linux or Red Hat Enterprise Linux, refer to the installation and configuration guides for that platform. For example, if you are installing Oracle RAC on the Oracle Solaris operating system, then you would use the following guides:

- *Oracle Grid Infrastructure Installation Guide for Oracle Solaris*
- *Oracle Real Application Clusters Installation Guide for Solaris Operating System*

Oracle Clusterware and Oracle RAC do not support heterogeneous platforms in the same cluster. For example, you cannot have one node in the cluster running Oracle Linux and another node in the same cluster running Oracle Solaris. All nodes must run the same operating system; that is, they must be binary compatible. Oracle RAC does not support machines having different chip architectures in the same cluster. However, you can have machines of different speeds and sizes in the same cluster.

See Also:

Oracle Real Application Clusters Administration and Deployment Guide for an overview of installing Oracle RAC

1.5 Tools for Installing, Configuring, and Managing Oracle RAC

There are different tools used for installing, configuring, and managing Oracle Clusterware, Oracle Automatic Storage Management (Oracle ASM), and Oracle Real Application Clusters (Oracle RAC).

Table 1-1 Tools for Installing, Configuring, and Managing Oracle RAC

Tool or Utility	Purpose	Description of Its Use
Oracle Universal Installer (OUI)	Installs Oracle Grid Infrastructure for a cluster (which consists of Oracle Clusterware and Oracle ASM) and Oracle Database with Oracle RAC.	After you configure the nodes to use in your cluster, OUI installs the Oracle software on the specified nodes using a network connection.

Table 1-1   (Cont.) Tools for Installing, Configuring, and Managing Oracle RAC

Tool or Utility	Purpose	Description of Its Use
Cluster Verification Utility (CVU)	Validates a range of cluster and Oracle RAC components such as shared storage devices, networking configurations, system requirements, and Oracle Clusterware, also operating system groups and users.	CVU is a command-line tool that you can use for preinstallation and postinstallation checks of your cluster environment. CVU is especially useful during preinstallation and during installation of Oracle Clusterware and Oracle RAC components. OUI runs CVU after the Oracle Clusterware installation to verify your environment.
ORAchk	Performs proactive heath checks for the Oracle software stack	ORAchk replaces the RACCheck utility, extends health check coverage to the entire Oracle software stack, and identifies and addresses top issues reported by Oracle users. ORAchk proactively scans for known problems with Oracle products and deployments.
Oracle Enterprise Manager	The primary product for managing your database	Oracle Enterprise Manager Cloud Control is a graphical user interface (GUI) that you use to manage single-instance and Oracle RAC databases, Oracle Clusterware, and Oracle ASM. After you have installed the Oracle Database software, created or upgraded a database, and configured the network, you can use Oracle Enterprise Manager to manage your database. Cloud Control also provides an interface for performance advisors and for Oracle utilities such as SQL*Loader and Recovery Manager (RMAN).
SQL*Plus	Issues SQL and PL/SQL statements to Oracle Database	SQL*Plus provides a command-line interface that enables you to perform database management operations for a database, and also to query, insert, update, or delete data directly in the database.

Table 1-1 (Cont.) Tools for Installing, Configuring, and Managing Oracle RAC

Tool or Utility	Purpose	Description of Its Use
Server Control (SRVCTL)	Manages the resources defined in the Oracle Cluster Registry (OCR). These resources include the node applications, called **nodeapps**, that comprise Oracle Clusterware, which includes the Oracle Notification Service (ONS), and the Virtual IP (VIP).	You use SRVCTL, a command-line interface, to manage databases, instances, listeners, services, and applications that run in an Oracle Clusterware environment. Using SRVCTL you can start and stop nodeapps, databases, instances, listeners, and services, delete or move instances and services, add services, and manage configuration information.
Cluster Ready Services Control (CRSCTL)	Manages Oracle Clusterware daemons. These daemons include Cluster Synchronization Services (CSS), Cluster-Ready Services (CRS), and Event Manager (EVM).	CRSCTL is a command-line tool that you can use to start and stop Oracle Clusterware and to determine the current status of your Oracle Clusterware installation.
Database Configuration Assistant (DBCA)	Creates and configures Oracle Databases	DBCA can be launched by OUI, depending upon the type of install that you select. You can also launch DBCA as a standalone tool at any time after Oracle Database installation. You can run DBCA in interactive mode or noninteractive/silent mode. Interactive mode provides a graphical interface and guided workflow for creating and configuring a database. DBCA is the preferred way to create a database, because it is a more automated approach, and your database is ready to use when DBCA completes.
Oracle Automatic Storage Management Configuration Assistant (ASMCA)	Installs and configures Oracle ASM	ASMCA is a utility that supports installing and configuring Oracle ASM instances, disk groups, volumes, and Oracle Automatic Storage Management Cluster File System (Oracle ACFS). ASMCA provides both a GUI and a non-GUI interface.

Table 1-1 (Cont.) Tools for Installing, Configuring, and Managing Oracle RAC

Tool or Utility	Purpose	Description of Its Use
Oracle Automatic Storage Management Command Line utility (ASMCMD)	Administers Oracle ASM	ASMCMD is a command-line utility that you can use to manage Oracle ASM instances, Oracle ASM disk groups, file access control for disk groups, files and directories within Oracle ASM disk groups, templates for disk groups, and Oracle ASM volumes.
Listener Control (LSNRCTL)	Administers listeners	The Listener Control utility is a command-line interface that you use to administer listeners. You can use its commands to perform basic management functions on one or more listeners. Additionally, you can view and change parameter settings for the listener.

If you have installed Oracle Grid Infrastructure for a cluster for Oracle Database 12*c* release 2 (12.2), then, when using utilities to manage your cluster, databases, database instances, Oracle ASM, and listeners, use the appropriate binary that is in the home directory of the object or component you are managing. Set the `ORACLE_HOME` environment variable to point to this directory, for example:

- If you use ASMCMD, SRVCTL, SQL*Plus, or LSNRCTL to manage Oracle ASM or its listener, then use the executable files located in the Grid home, not the executable files located in the Oracle Database home, and set the `ORACLE_HOME` environment variable to the location of the Grid home.
- If you use SRVCTL, SQL*Plus, or LSNRCTL to manage a database instance or its listener, then use the binaries located in the Oracle home where the database instance or listener is running, and set the `ORACLE_HOME` environment variable to the location of that Oracle home

See Also:

Oracle Real Application Clusters Administration and Deployment Guide for more information about the tools and utilities used to manage Oracle RAC

2
Preparing Your Cluster

This chapter contains the information that your system administrator and network administrator need to help you, as the DBA, configure two nodes in your cluster. This chapter assumes a basic understanding of the Linux operating system. In some cases, you may need to refer to details in *Oracle Grid Infrastructure Installation and Upgrade Guide* for your operating system. In addition, you must have these additional privileges have to perform certain tasks in this chapter: `root` or `sudo` privileges for UNIX and Linux systems, Administrator privileges on Windows systems.

- Verifying System Requirements
  Before you begin your installation, check to ensure that your system meets the requirements for Oracle Real Application Clusters (Oracle RAC).
- Preparing the Server
  After you have verified that your system meets the basic requirements for installing Oracle RAC, the next step is to configure the server in preparation for installation.
- Configuring the Network
  Before you install Oracle Grid Infrastructure and Oracle RAC you need to decide on network names and configure IP addresses.
- Preparing the Operating System and Software
  When you install the Oracle software on your server, Oracle Universal Installer expects the operating system to have specific packages and software applications installed.
- Configuring Installation Directories and Shared Storage
  Oracle RAC requires access to a shared file system for storing Oracle Clusterware files. You must also determine where the Oracle software and database files will be installed.

> **See Also:**
>
> - "About Installing Oracle RAC on Different Operating Systems"
> - "Tools for Installing, Configuring, and Managing Oracle RAC"

2.1 Verifying System Requirements

Before you begin your installation, check to ensure that your system meets the requirements for Oracle Real Application Clusters (Oracle RAC).

The requirements can be grouped into the following categories:

- Checking Operating System Certifications
  You must ensure that you have a certified combination of the operating system and the Oracle Database software.

- About Hardware Requirements
  Each node you make part of your cluster, or Oracle Clusterware and Oracle RAC installation, must satisfy the minimum hardware requirements of the software.
- About Shared Storage
  Oracle Grid Infrastructure and Oracle RAC require access to shared storage, so each node and instance in the cluster can access the same set of files.
- About Network Hardware Requirements
  You must have the hardware to support communication, both public and private, between all the nodes in the cluster.
- About IP Address Requirements
  Oracle Grid Infrastructure and Oracle RAC use a variety of IP addresses for communication.
- Verifying Operating System and Software Requirements
  Ensure your system meets the operating system version and other software requirements.

2.1.1 Checking Operating System Certifications

You must ensure that you have a certified combination of the operating system and the Oracle Database software.

For a list of the certified operating systems and Oracle Database software, refer to My Oracle Support certification, which is located at the following website

`https://support.oracle.com`

You can find certification information by selecting the **Certifications** tab. You can also search for support notes that contain instructions on how to locate the Certification information for your platform.

> **Note:**
>
> Oracle Universal Installer verifies that your server and operating system meet the listed requirements. However, you should check the requirements before you start Oracle Universal Installer to ensure your server and operating system meet the requirement. This helps to avoid delays in the software installation process that you might incur if your hardware or software is not certified.

> **See Also:**
>
> - "Preparing the Server"
> - "Verifying Operating System and Software Requirements"

2.1.2 About Hardware Requirements

Each node you make part of your cluster, or Oracle Clusterware and Oracle RAC installation, must satisfy the minimum hardware requirements of the software.

> **Note:**
>
> When you install Oracle software, Oracle Universal Installer (OUI) automatically performs hardware prerequisite checks and notifies you if they are not met.

The minimum hardware requirements are:

- At least 4 GB of RAM for Oracle Grid Infrastructure for a Cluster installations, including installations where you plan to install Oracle RAC
- An amount of swap space that is at least equal to the amount of RAM
- At least two network switches, each at least 1 GbE, to support the public and private networks
- Servers should be either in runlevel 3 or runlevel 5.
- Temporary space (at least 1 GB) available in `/tmp`
- A processor type (CPU) that is certified with the release of the Oracle software being installed (64-bit)
- A minimum of 1024 x 786 display resolution, so that Oracle Universal Installer (OUI) displays correctly
- All servers in the cluster have the same chip architecture, for example, all 64-bit processors
- Access to either Storage Area Network (SAN) or Network-Attached Storage (NAS).

You must also have sufficient disk space in the software installation locations to store the Oracle software, as described in the following table.

Location	Amount	Purpose
Grid home directory	At least 8 GB	Software installation of Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM)
Grid home directory	100 GB is recommended	Additional disk space for the associated log files and patches
Oracle base of the Oracle Grid Infrastructure installation owner (Grid user)	At least 10 GB	Oracle Clusterware and Oracle ASM log files and for diagnostic collections generated by Trace File Analyzer (TFA) Collector
Oracle home	At least 6.4 GB	Oracle Database software binaries

Note:

Refer to the *Oracle Grid Infrastructure Installation and Upgrade Guide* and the *Oracle Real Application Clusters Installation Guide* for your operating system for the actual disk space requirements. The amount of disk space used by the Oracle software can vary, and might be higher than what is listed in this guide.

See Also:

- "About Performing Platform-Specific Configuration Tasks"
- "Preparing the Server"
- "Configuring Installation Directories and Shared Storage "

2.1.3 About Shared Storage

Oracle Grid Infrastructure and Oracle RAC require access to shared storage, so each node and instance in the cluster can access the same set of files.

An Oracle RAC database is a *shared everything* database. All data files, control files, redo log files, and the server parameter file (SPFILE) used by the Oracle RAC database must reside on shared storage that is accessible by all the Oracle RAC database instances. The Oracle RAC installation demonstrated in this guide uses Oracle ASM for the shared storage for Oracle Clusterware and Oracle Database files.

Oracle Clusterware achieves superior scalability and high availability by using the following components:

- **Voting disk**–Manages cluster membership and arbitrates cluster ownership between the nodes in case of network failures. The voting disk is a file that resides on shared storage. For high availability, Oracle recommends that you have multiple voting disks, and that you have an odd number of voting disks. If you define a single voting disk, then use mirroring at the file system level for redundancy.
- **Oracle Cluster Registry (OCR)**–Maintains cluster configuration information and configuration information about any cluster database within the cluster. The OCR contains information such as which database instances run on which nodes and which services run on which databases. The OCR also stores information about processes that Oracle Clusterware controls. The OCR resides on shared storage that is accessible by all the nodes in your cluster. Oracle Clusterware can multiplex, or maintain multiple copies of, the OCR and Oracle recommends that you use this feature to ensure high availability.
- **Grid Infrastructure Management Repository**–Supports Oracle Database Quality of Service management, Memory Guard, and Cluster Health Monitor

These Oracle Clusterware components require the following disk space on a shared file system:

- Three Oracle Clusterware Registry (OCR) files, a least 400 MB for each volume, or 1.2 GB total disk space
- Three voting disk files, 300 MB each volume, or 900 MB total disk space
- The Grid Infrastructure Management Repository requires an additional 5.9 GB of disk space and is stored in an OCR volume

> **See Also:**
>
> - The chapter on configuring storage in *Oracle Grid Infrastructure Installation and Upgrade Guide*
> - "Configuring Installation Directories and Shared Storage "
> - "Configuring Shared Storage"

2.1.4 About Network Hardware Requirements

You must have the hardware to support communication, both public and private, between all the nodes in the cluster.

Oracle Clusterware requires that you connect the nodes in the cluster to a private network by way of a private interconnect . The private interconnect is a separate network that you configure between cluster nodes. The interconnect serves as the communication path between nodes in the cluster. This interconnect should be a *private* interconnect, meaning it is *not* accessible to nodes that are not members of the cluster.

The interconnect used by Oracle RAC is the same interconnect that Oracle Clusterware uses. Each cluster database instance uses the interconnect for messaging to synchronize the use of shared resources by each instance. Oracle RAC also uses the interconnect to transmit data blocks that are shared between the instances.

When you configure the network for Oracle RAC and Oracle Clusterware, each node in the cluster must meet the following requirements:

- Each node must have at least two network interface cards (NICs), or **network adapters**. One adapter is for the public network interface and the other adapter is for the private network interface (the interconnect). Install additional network adapters on a node if that node meets either of the following conditions:
  - Does not have at least two network adapters
  - Has two network interface cards but is using network attached storage (NAS). You should have a separate network adapter for NAS.
  - Has two network cards, but you want to use the **Redundant Interconnect Usage** feature, which allows multiple network adapters to be accessed as a single adapter. Redundant Interconnect Usage enables load-balancing and high availability across multiple (up to 4) private networks (also known as interconnects).
- For easier management, public interface names should be the same for all nodes. If the public interface on one node uses the network adapter `eth0`, then you should

configure `eth0` as the public interface on all nodes. Network interface names are case-sensitive.

- The network adapter for the public interface must support TCP/IP.
- The network adapter for the private interface must support the user datagram protocol (UDP) using high-speed network adapters and a network switch that supports TCP/IP (Gigabit Ethernet or better).

> **Note:**
>
> - UDP is the default interface protocol for Oracle RAC and Oracle Clusterware.
> - You must use a switch for the interconnect. Oracle recommends that you use a dedicated network switch. Token-rings or crossover cables are not supported for the interconnect.
> - Loopback devices are not supported.

- For the private network, the endpoints of all designated interconnect interfaces must be completely reachable on the network. Every node in the cluster must be able to connect to every private network interface in the cluster.
- The host name of each node must conform to the RFC 952 standard, which permits alphanumeric characters. Host names using underscores ("_") are not allowed.

> **See Also:**
>
> - *Oracle Grid Infrastructure Installation and Upgrade Guide for Linux* for more information about the configuration requirements for Redundant Interconnect Usage
> - "Configuring the Network"
> - "Verifying System Requirements"

2.1.5 About IP Address Requirements

Oracle Grid Infrastructure and Oracle RAC use a variety of IP addresses for communication.

When performing an advanced installation of Oracle Grid Infrastructure for a cluster, you can chose to use Grid Naming Service (GNS) and Dynamic Host Configuration Protocol (DHCP) for virtual IPs (VIPs). GNS uses multicast Domain Name Server (mDNS) to enable the cluster to assign host names and IP addresses dynamically as nodes are added and removed from the cluster, without requiring additional network address configuration in the domain name server (DNS).

You can configure the public network interfaces for cluster nodes to use IPv4, IPv6, or both types of IP addresses. During installation, the nodes can be connected to networks that have both address types, but you cannot configure the VIPs to use both address types during installation. After installation, you can configure cluster member nodes with a mixture of IPv4 and IPv6 addresses. Oracle Grid Infrastructure and Oracle RAC support the standard IPv6 address notations specified by RFC 2732.

This guide documents how to perform a typical installation, which does not use GNS and uses only IPv4 addresses. You must configure the following addresses manually in your corporate DNS:

- A fixed public IP address for each node
- A virtual IP address for each node
- Three single client access name (SCAN) addresses for the cluster

> **Note:**
>
> Oracle Clusterware uses interfaces marked as private as the cluster interconnects. You do not need to specify IP addresses for the interconnect.

During installation a SCAN for the cluster is configured, which is a domain name that resolves to all the SCAN addresses allocated for the cluster. The public IP addresses, SCAN addresses, and VIP addresses must all be on the same subnet. The SCAN must be unique within your network. The SCAN addresses should not respond to `ping` commands before installation. In a Typical installation, the SCAN is also the name of the cluster, so the SCAN must meet the same requirements as the cluster name and can be no longer than 15 characters.

During installation of the Oracle Grid Infrastructure for a cluster, a listener is created for each of the SCAN addresses. Clients that access the Oracle RAC database should use the SCAN or SCAN address, not the VIP name or address. If an application uses a SCAN to connect to the cluster database, then the network configuration files on the client computer do not have to be modified when nodes are added to or removed from the cluster. The SCAN and its associated IP addresses provide a stable name for clients to use for connections, independent of the nodes that form the cluster. Clients can connect to the cluster database using the easy connect naming method and the SCAN.

> **See Also:**
>
> - "Configuring the Network"
> - "Verifying System Requirements"
> - *Oracle Database Net Services Administrator's Guide* for information about the easy connect naming method
> - *Oracle Grid Infrastructure Installation and Upgrade Guide* for your platform for more information about configuring GNS and IPv6 network addresses for your cluster

2.1.6 Verifying Operating System and Software Requirements

Ensure your system meets the operating system version and other software requirements.

Refer to the *Oracle Grid Infrastructure Installation and Upgrade Guide* and the *Oracle Real Application Clusters Installation Guide* for your platform for information about specific requirements for your environment.

- About Operating System and Software Requirements
  Before installing the Oracle software, check that your operating system meets the requirements.
- About Installation Fixup Scripts
  Oracle Universal Installer (OUI) detects when the minimum requirements for an installation are not met, and creates shell scripts, called Fixup scripts, to finish incomplete system configuration steps.
- Checking the Current Operating System Configuration
  Instead of waiting for the installation to notify you of an incorrect configuration, you can use operating system commands to manually check the operating system configuration before installation. This helps you determine if additional time will be needed to update the operating system before starting the Oracle Grid Infrastructure installation.

2.1.6.1 About Operating System and Software Requirements

Before installing the Oracle software, check that your operating system meets the requirements.

The operating system and software requirements might include:

- The operating system release
- The kernel version of the operating system
- Modifying the values for kernel parameters
- Installed packages, patches, or patch sets
- Installed compilers and drivers
- Web browser type and version
- Additional application software requirements

If you are currently running an operating system release that is not supported by Oracle Database 12*c* release 2 (12.2), then you must first upgrade your operating system before installing Oracle Real Application Clusters.

If you are using Oracle Linux as your operating system, then you can use the Oracle Validated RPM system configuration script to configure your system.

2.1.6.2 About Installation Fixup Scripts

Oracle Universal Installer (OUI) detects when the minimum requirements for an installation are not met, and creates shell scripts, called Fixup scripts, to finish incomplete system configuration steps.

If OUI detects an incomplete task, then it generates a Fixup script (`runfixup.sh`). You can run the script after you click **Fix and Check Again**.

Fixup scripts do the following:

- If necessary, set kernel parameters to values required for successful installation, including:
  - Shared memory parameters
  - Open file descriptor and UDP send/receive parameters
- Create and set permissions on the Oracle Inventory (central inventory) directory.
- Create or reconfigure primary and secondary group memberships for the installation owner, if necessary, for the Oracle Inventory directory and the operating system privileges groups
- Set shell limits, if necessary, to required values.

2.1.6.3 Checking the Current Operating System Configuration

Instead of waiting for the installation to notify you of an incorrect configuration, you can use operating system commands to manually check the operating system configuration before installation. This helps you determine if additional time will be needed to update the operating system before starting the Oracle Grid Infrastructure installation.

To determine if the operating system requirements for Oracle Linux have been met:

1. To determine which distribution and version of Linux is installed, enter the one of the following commands at the operating system prompt as the `root` user:

   ```
   # cat /etc/oracle-release
   # cat /etc/redhat-release
   # lsb_release -id
   ```

2. To determine which chip architecture each server is using and which version of the software you should install, run the following command at the operating system prompt as the `root` user:

   ```
   # uname -m
   ```

   This command displays the processor type. For a 64-bit architecture, the output would be "x86_64".

3. To determine if the required errata level is installed, run the `uname` command as the `root` user, as shown in this example:

   ```
   # uname -r
   2.6.39-100.7.1.el6uek.x86_64
   ```

   Like most software, the Linux kernel is updated to fix bugs in the operating system. These kernel updates are referred to as erratum kernels or *errata levels*.

   The output in the previous example shows that the kernel version is 2.6.39, and the errata level (EL) is 100.7.1.e16uek. Review the required errata level for your distribution. If the errata level is below the required minimum errata level, then install the latest kernel update for your operating system. The kernel updates are available from your operating system vendor.

4. To ensure there are no operating system issues affecting installation, make sure you have installed all the operating system patch updates and packages that are listed in *Oracle Grid Infrastructure Installation Guide* for your platform. If you are using Oracle Linux, then you can determine if the required *packages*, or programs that perform specific functions or calculations, are installed by using the following command as the `root` user:

   ```
   # rpm -q package_name
   ```

   The variable *`package_name`* is the name of the package you are verifying, such as `setarch`. If a package is not installed, then install it from your Linux distribution media or download the required package version from your Linux vendor's website.

   You can also use either `up2date` or YUM (Yellow dog Updater Modified) to install packages and their dependencies on some Linux systems. YUM uses repositories to automatically locate and obtain the correct RPM packages for your system.

> **See Also:**
>
> - "About Installing Oracle RAC on Different Operating Systems"
> - "Preparing the Server"
> - "Preparing the Operating System and Software"
> - "About Configuring the Software Owner's Shell Environment on Linux Systems"
> - "About Performing Platform-Specific Configuration Tasks"
> - *Oracle Grid Infrastructure Installation Guide* and the *Oracle Real Application Clusters Installation Guide* for your platform

2.2 Preparing the Server

After you have verified that your system meets the basic requirements for installing Oracle RAC, the next step is to configure the server in preparation for installation.

- About Operating System Users and Groups
  Depending on whether this is the first time Oracle software is installed on this server, you may have to create operating system groups and users.
- Configuring Operating System Users and Groups on Linux Systems
  This task describes how to create the `grid` and `oracle` users prior to installing the software.
- Configuring Secure Shell on Linux Systems
  To install Oracle software, Secure Shell (SSH) connectivity must be set up between all cluster member nodes.
- About Configuring the Software Owner's Shell Environment on Linux Systems
  On Oracle Linux, you run Oracle Universal Installer (OUI) from the `grid` account. OUI obtains information from the environment variables configured for the `grid` user.

See Also:

"About Performing Platform-Specific Configuration Tasks"

2.2.1 About Operating System Users and Groups

Depending on whether this is the first time Oracle software is installed on this server, you may have to create operating system groups and users.

- Required Operating System Users and Groups
  To install Oracle Grid Infrastructure for a cluster and Oracle Real Application Clusters (Oracle RAC), you must create certain operating system groups and users.
- Separate Operating System Users and Groups for Oracle Software Installations on UNIX and Linux Systems
  Instead of using a single operating system user as the owner of every Oracle software installation, you can use multiple users, each owning one or more Oracle software installations.
- Separate Operating System Users and Groups for Oracle Software Installations on Windows Systems
  On the Windows platform, management of operating system users and groups is different from Linux and UNIX Systems.
- Optional Operating System Users and Groups
  You can use additional users and groups to divide administrative access privileges to the Oracle Grid Infrastructure for a cluster installation from other administrative users and groups associated with other Oracle installations.

2.2.1.1 Required Operating System Users and Groups

To install Oracle Grid Infrastructure for a cluster and Oracle Real Application Clusters (Oracle RAC), you must create certain operating system groups and users.

Table 2-1 Required Operating System Users and Groups for Oracle Grid Infrastructure and Oracle RAC Installations

Operating System User or Group	Description
The Oracle Inventory group for all Oracle software installations (`oinstall`).	The Oracle Inventory group must be the primary group for Oracle software installation owners on Linux and UNIX platforms. Members of the Oracle Inventory group have access to the Oracle Inventory directory. This directory is the central inventory record of all Oracle software installations on a server and the installation logs and trace files from each installation. On Windows systems, this group is created and managed automatically.

Table 2-1 (Cont.) Required Operating System Users and Groups for Oracle Grid Infrastructure and Oracle RAC Installations

Operating System User or Group	Description
An Oracle software owner, or installation user. This is the user account you use when installing the software.	If you want to use a single software owner for all installations, then typically the user name is `oracle`. If you plan to install Oracle Grid Infrastructure for a cluster and Oracle RAC using separate software owners to separate Oracle Grid Infrastructure for a cluster administrative privileges from Oracle Database administrative privileges, then typically you would use `grid` for Oracle Grid Infrastructure for a cluster owner and `oracle` for the Oracle RAC owner. Using separate users to install Oracle Grid Infrastructure and Oracle RAC is the recommended configuration.
The OSRAC group for Oracle Database authentication.	The OSRAC group is a system privileges group whose members are granted the SYSRAC privilege to administer Oracle Database and the SYSASM privilege to administer Oracle Clusterware and Oracle ASM. To provide fine-grained control of administrative privileges you can create multiple operating system groups as described in "Optional Operating System Users and Groups".

Note:

When installing Oracle RAC on a Microsoft Windows platform:

- OUI automatically creates the groups used for system authenticating access to the Oracle software.
- The user performing the installation must be an Administrator user.
- If you specify an Oracle Home user during installation, this user must be a domain user.

On Linux and UNIX systems, if you use one installation owner for both Oracle Grid Infrastructure for a cluster and Oracle RAC, then when you want to perform administration tasks, you must change the value for `ORACLE_HOME` environment variable to match the instance you want to administer (Oracle ASM, in the Grid home, or a database instance in the Oracle home). To change the `ORACLE_HOME` environment variable, use a command syntax similar to the following example, where `/u01/app/12.2.0/grid` is the Oracle Grid Infrastructure for a cluster home:

```
ORACLE_HOME=/u01/app/12.2.0/grid; export ORACLE_HOME
```

If you try to administer an instance using SQL*Plus, LSNRCTL, or ASMCMD while `ORACLE_HOME` is set to an incorrect binary path, then you will encounter errors. The Oracle home path does not affect SRVCTL commands.

On Windows systems, to perform administration tasks, you simply run the appropriate utility for either Oracle Grid Infrastructure or Oracle RAC and ORACLE_HOME is configured automatically.

2.2.1.2 Separate Operating System Users and Groups for Oracle Software Installations on UNIX and Linux Systems

Instead of using a single operating system user as the owner of every Oracle software installation, you can use multiple users, each owning one or more Oracle software installations.

Instead of using a single operating system user as the owner of every Oracle software installation, you can use multiple users, each owning one or more Oracle software installations. A user created to own only the Oracle Grid Infrastructure for a cluster installation is called the grid user. This user owns both the Oracle Clusterware and Oracle Automatic Storage Management binaries. A user created to own either all Oracle software installations (including Oracle Grid Infrastructure for a cluster), or only Oracle Database software installations, is called the oracle user.

You can also use different users for each Oracle Database software installation. Additionally, you can specify a different OSDBA group for each Oracle Database software installation. By using different operating system groups for authenticating administrative access to each Oracle Database installation, users have SYSDBA or SYSRAC privileges for the databases associated with their OSDBA or OSRAC group, rather than for all the databases on the system.

Members of the OSDBA group can also be granted the SYSASM system privilege, which gives them administrative access to Oracle ASM. As described in "Optional Operating System Users and Groups", you can configure a separate operating system group for Oracle ASM authentication to separate users with SYSASM access to the Oracle ASM instances from users with SYSDBA access to the database instances.

If you want to create separate Oracle software owners so you can use separate users and operating system privilege groups for the different Oracle software installations, then note that each of these users must have the Oracle central inventory group (oinstall) as their primary group. Members of this group have the required write privileges to the Oracle Inventory directory.

> **Note:**
>
> The Oracle Grid Infrastructure for a cluster installation can be owned by only one user. You cannot have one user that owns the Oracle Clusterware installed files and a different user that owns the Oracle ASM installed files.

2.2.1.3 Separate Operating System Users and Groups for Oracle Software Installations on Windows Systems

On the Windows platform, management of operating system users and groups is different from Linux and UNIX Systems.

You create at least one Oracle Installation user (an Administrator user who installs Oracle software) when you install Oracle Grid Infrastructure. You can use this Oracle

Installation user or a different Oracle Installation user when installing Oracle Database software.

You should create additional non-privileged Windows user accounts to use as Oracle Home users during installation. An Oracle Home User is different from an Oracle Installation user. An Oracle Installation user is the user who needs administrative privileges to install Oracle products. An Oracle Home user is a low-privileged Windows User Account specified during installation that runs most of the Windows services required for the Oracle home. Different Oracle homes on a system can share the same Oracle Home user or use different Oracle Home users. You do not need to create any operating system groups because they are created and managed automatically by the Oracle software.

The Oracle Home user can be a Built-in Account or a Windows domain user. The Windows domain user should be a low-privileged, non-administrative account to ensure that the Oracle Database services running under Oracle Home User have only those privileges required to run Oracle products. Create separate Oracle Home users for each Oracle software product to have separate administrative controls for the different Oracle Homes.

If you use different Oracle Home users for each Oracle Database software installation, then you can use the `ORA_`*`HOMENAME`*`_RAC` group that is associated with each Oracle home to separate SYSRAC privileges for each installation. A member of an `ORA_`*`HOMENAME`*`_RAC` group can use operating system authentication to log in to only the Oracle Databases that run from the Oracle home with the name *`HOMENAME`*. Members of the `ORA_RAC` group can use operating system authentication to log in to any Oracle database with SYSRAC privileges.

You can also use the Oracle ASM access control feature to enable role separation for Oracle ASM management. In previous releases, this feature was disabled on Windows because all Oracle Database services ran as Local System. For Oracle ASM administration, the new groups `ORA_ASMADMIN`, `ORA_ASMDBA` and `ORA_ASMOPER` are automatically created and populated during Oracle Grid Infrastructure installation.

2.2.1.4 Optional Operating System Users and Groups

You can use additional users and groups to divide administrative access privileges to the Oracle Grid Infrastructure for a cluster installation from other administrative users and groups associated with other Oracle installations.

You implement separate administrative access for users by specifying membership in different operating system groups, You implement separate installation privileges by using different installation owners for each Oracle installation. The optional users and groups you can use are:

- The Oracle Automatic Storage Management Group, or OSASM group (typically `asmadmin` for Linux and `ORA_ASMADMIN` for Windows).
- The ASM Database Administrator group, or OSDBA for ASM group (typically `asmdba` for Linux and ORA_ASMDBA for Windows).
- The OSOPER for Oracle Database group (typically, `oper` for Linux and `ORA_OPER` for Windows).
- The OSOPER for Oracle ASM group (typically `asmoper` for Linux and `ORA_ASMOPER` for Windows).
- The OSRACDBA group for the cluster agents to use the SYSRAC role instead of the SYSDBA role (typically `racdba` for Linux and `ORA_RAC` for Windows).

- The OSBACKUPDBA group for Oracle Database.
- The OSDGDBA group for Oracle Data Guard.
- The OSKMDBA group for encryption key management.

> **See Also:**
>
> - About Configuring the Software Owner's Shell Environment on Linux Systems
> - *Oracle Database Security Guide* for information about the privileges associated with administrative users and groups
>
>   *Oracle Grid Infrastructure Installation and Upgrade Guide* for your operating system for more information about configuring separation of privileges by role.

2.2.2 Configuring Operating System Users and Groups on Linux Systems

This task describes how to create the `grid` and `oracle` users prior to installing the software.

The instructions in this guide use one software owner for the Oracle Grid Infrastructure installation, and a separate user is used for the Oracle Database (Oracle Real Application Clusters) installation. The users are named `grid` and `oracle`. The `oracle` user belongs to the `oinstall` and `dba` operating system groups.

To create software owners with all operating system-authenticated administration privileges:

1. Determine the groups that exist on your server by listing the contents of the `/etc/group` file.

   ```
   cat /etc/group
   ```

2. If this is the first time Oracle software has been installed on your server, and the Oracle Inventory group does not exist, then create the Oracle Inventory group (`oinstall`) with a group ID that is currently not in use on all the nodes in your cluster. Enter a command as the `root` user that is similar to the following:

   ```
   # /usr/sbin/groupadd -g 1000 oinstall
   ```

3. Create an OSDBA (`dba`) group with a group ID that is currently not in use on all the nodes in your cluster by entering a command as the `root` user that is similar to the following:

   ```
   # /usr/sbin/groupadd -g 1001 dba
   ```

4. Create an OSDBA group for Oracle ASM (`asmdba`) with a group ID that is currently not in use on all the nodes in your cluster by entering a command as the `root` user that is similar to the following:

   ```
   # /usr/sbin/groupadd -g 1002 asmdba
   ```

5. If the user that owns the Oracle software (`grid` or `oracle`) does not exist on your server, then you must create the user. Select a user ID (UID) that is currently not in use on all the nodes in your cluster. To determine which users have been created on your server, list the contents of the `/etc/passwd` file using the command:

   ```
   cat /etc/passwd
   ```

   The following command shows how to create the `grid` user and the user's home directory (`/home/grid`) with the default group as `oinstall` and the secondary groups as `dba` and `asmdba`, using a UID of 1100:

   ```
   # /usr/sbin/useradd -u 1100 –g oinstall -G dba,asmdba -d /home/grid -r grid
   ```

   The following command shows how to create the `oracle` user and the user's home directory (`/home/oracle`) with the default group as `oinstall` and the secondary group as `dba`, using a UID of 1100:

   ```
   # /usr/sbin/useradd -u 1200 –g oinstall -G dba,asmdba -d /home/oracle -r oracle
   ```

6. Set the password for the `grid` and `oracle` accounts using the following command. Replace *`password`* with your own password.

   ```
   passwd grid
   Changing password for user grid.
   New UNIX password: password
   retype new UNIX password: password
   passwd:  all authentication tokens updated successfully.
   passwd oracle
   Changing password for user oracle.
   New UNIX password: password
   retype new UNIX password: password
   passwd:  all authentication tokens updated successfully.
   ```

7. Repeat Step 1 through Step 6 on each node in your cluster.

8. Verify that the attributes of the user `grid` and `oracle` are identical on each node of your cluster:

   ```
   id grid
   id oracle
   ```

   The command output should be similar to the following:

   ```
   uid=1100(grid) gid=1000(oinstall) groups=1000(oinstall),1001(dba),1002(asmdba)
   uid=1200(oracle) gid=1000(oinstall) groups=1000(oinstall),1001(dba),1002(asmdba)
   ```

> **See Also:**
>
> - "Configuring Installation Directories and Shared Storage "
> - "About Oracle Automatic Storage Management"

2.2.3 Configuring Secure Shell on Linux Systems

To install Oracle software, Secure Shell (SSH) connectivity must be set up between all cluster member nodes.

OUI uses the ssh and scp commands during installation to run remote commands on and copy files to the other cluster nodes. You must configure SSH so that these commands do not prompt for a password. SSH is also used by the configuration assistants, Enterprise Manager, and when adding nodes to the cluster.

You can configure SSH from the Oracle Universal Installer (OUI) interface during installation for the user account running the installation. The automatic configuration creates passwordless SSH connectivity between all cluster member nodes.

To enable the script to run, you must remove stty commands from the profiles of any Oracle software installation owners, and remove other security measures that are triggered during a login, and that generate messages to the terminal. These messages, mail checks, and other displays prevent Oracle software installation owners from using the SSH configuration script that is built into the Oracle Universal Installer. If they are not disabled, then SSH must be configured manually before an installation can be run.

On Linux systems, to enable Oracle Universal Installer to use the ssh and scp commands without being prompted for a pass phrase, you must have user equivalency in the cluster. User equivalency exists in a cluster when the following conditions exist on *all* nodes in the cluster:

- A given user has the same user name, user ID (UID), and password.
- A given user belongs to the same groups.
- A given group has the same group ID (GID).

See Also:

- "Configuring Operating System Users and Groups on Linux Systems" for information about configuring user equivalency
- *Oracle Grid Infrastructure Installation Guide* for your platform for more information about manually configuring SSH

2.2.4 About Configuring the Software Owner's Shell Environment on Linux Systems

On Oracle Linux, you run Oracle Universal Installer (OUI) from the grid account. OUI obtains information from the environment variables configured for the grid user.

Before running OUI, you must make the following changes to the shell startup file for the software owner of Oracle Grid Infrastructure for a cluster:

- Set the default file mode creation mask (umask) of the installation user (grid) to 022 in the shell startup file. Setting the mask to 022 ensures that the user performing the software installation creates files with 644 permissions.
- Set ulimit settings for file descriptors (nofile) and processes (nproc) for the installation user (grid).
- Set the software owner's environment variable DISPLAY environment variables in preparation for the Oracle Grid Infrastructure for a cluster installation.

- Remove any lines in the file that set values for `ORACLE_SID`, `ORACLE_HOME`, or `ORACLE_BASE` environment variables.

After you have saved your changes, run the shell startup script to configure the environment.

Also, if the `/tmp` directory has less than 1 GB of available disk space, but you have identified a different, non-shared file system that has at least 1 GB of available space, then you can set the `TEMP` and `TMPDIR` environment variables to specify the alternate temporary directory on this file system.

To review your current environment settings, use the `env | more` command as the `grid` user.

Note:

Remove any `stty` commands from hidden files (such as logon or profile scripts) before you start the installation. On Linux systems, if there are any such files that contain `stty` commands, then when these files are loaded by the remote shell during installation, OUI indicates an error and stops the installation.

See Also:

- *Oracle Grid Infrastructure Installation Guide* for your platform for more information on how to configure the Oracle software owner environment before installation
- "About Operating System Users and Groups"

2.3 Configuring the Network

Before you install Oracle Grid Infrastructure and Oracle RAC you need to decide on network names and configure IP addresses.

Clients and applications connect to the database using a single client access name (SCAN). The SCAN and its associated IP addresses provide a stable name for clients to use for connections, independent of the nodes that form the cluster. A SCAN works by being able to resolve to multiple IP addresses referencing multiple listeners in the cluster that handle public client connections. The installation process creates these listeners, called SCAN listeners.

To configure the network in preparation for installing Oracle Grid Infrastructure for a cluster:

1. Determine your cluster name.

   The cluster name should satisfy the following conditions:

   - The cluster name is at least one character long and less than 15 characters long.

- The cluster name must consist of the same character set used for host names: single-byte alphanumeric characters (a to z, A to Z, and 0 to 9) and hyphens (-).
- The cluster name is globally unique throughout your host domain
- If you use third-party vendor clusterware, then Oracle recommends that you use the vendor cluster name.

2. Determine the public host name and virtual host name for each node in the cluster.
   - For the public host name, use the primary host name of each node. In other words, use the name displayed by the `hostname` command. This host name can be either the permanent or the virtual host name, for example: `racnode1`.
   - Determine a virtual host name for each node. A virtual host name is a public node name that Oracle Clusterware uses to reroute client requests sent to the node if the node is down. Oracle recommends that you provide a name in the format <*public hostname*>`-vip`, for example: `racnode1-vip`.
3. Identify the interface names and associated IP addresses for all network adapters by executing the following command on each node:

   ```
   # /sbin/ifconfig
   ```

   From the output, identify the interface name (such as `eth0`) and IP address for each network adapter you specify as a public network interface.

   If your operating system supports a graphic user interface (GUI) for modifying the system configuration, then you can also use the following command to start a GUI that you can use to configure the network adapters and `/etc/hosts` file:

   ```
   /usr/bin/system-config-network &
   ```

   Note:

   When you install Oracle Clusterware and Oracle RAC, you are asked to provide the interface name and IP address for each network adapter.

4. On each node in the cluster, assign a public IP address with an associated network name to one network adapter. The public name for each node should be registered with your domain name system (DNS). IP addresses on the subnet you identify as private are assigned as private IP addresses for cluster member nodes. You do not have to configure these addresses manually in the `/etc/hosts` file.

   You can test whether an interconnect interface is reachable using a `ping` command.

5. On each node in the cluster, configure a third IP address that serves as a virtual IP address.

   Use an IP address that meets the following requirements:

   - The virtual IP address and the network name must *not* be currently in use.
   - The virtual IP address must be on the *same* subnet as your public IP address.

   The virtual host name for each node should be registered with your DNS.

6. Define a SCAN that resolves to three IP addresses in your DNS.

7. When you complete the network configuration, the IP address and network interface configuration should be similar to what is shown in the following table (Your node names and IP addresses might be different):

Table 2-2 Manual Network Configuration Example

Identity	Home Node	Host Node	Given Name	Type	Address	Address Assigned By	Resolved By
Node 1 Public	Node 1	`racnode1`	`racnode1`[1]	Public	192.0.2.101	Fixed	DNS
Node 1 VIP	Node 1	Selected by Oracle Clusterware	`racnode1-vip`	Virtual	192.0.2.104	Fixed	DNS, hosts file
Node 1 Private	Node 1	`racnode1`	`racnode1-priv`	Private	192.168.0.1	Fixed	DNS, hosts file, or none
Node 2 Public	Node 2	`racnode2`	`racnode2`[1]	Public	192.0.2.102	Fixed	DNS
Node 2 VIP	Node 2	Selected by Oracle Clusterware	`racnode2-vip`	Virtual	192.0.2.105	Fixed	DNS, hosts file
Node 2 Private	Node 2	`racnode2`	`racnode2-priv`	Private	192.168.0.2	Fixed	DNS, hosts file, or none
SCAN VIP 1	none	Selected by Oracle Clusterware	`docrac-scan`	virtual	192.0.2.201	Fixed	DNS
SCAN VIP 2	none	Selected by Oracle Clusterware	`docrac-scan`	virtual	192.0.2.202	Fixed	DNS
SCAN VIP 3	none	Selected by Oracle Clusterware	`docrac-scan`	virtual	192.0.2.203	Fixed	DNS

[1] Node host names may resolve to multiple addresses.

Even if you are using a DNS, Oracle recommends that you add lines to the `/etc/hosts` file on each node, specifying the public IP addresses. Configure the `/etc/hosts` file so that it is similar to the following example:

```
#eth0 - PUBLIC
192.0.2.100       racnode1.example.com       racnode1
192.0.2.101       racnode2.example.com       racnode2
```

> **Note:**
>
> Make a note of the addresses you configured in the `/etc/hosts` file or registered with DNS. When you install Oracle Grid Infrastructure for a cluster and Oracle RAC, you are prompted for the public, virtual IP, and SCAN addresses.
>
> Oracle strongly recommends that you do not configure SCAN VIP addresses in the `hosts` file. Use DNS resolution for SCAN VIPs.

The fully qualified SCAN for the cluster defaults to *`cluster_name`*`-scan.`*`GNS_subdomain_name`*, for example `docrac-scan.grid.example.com`. The short SCAN for the cluster is `docrac-scan`. Without GNS, and configuring SCAN in DNS directly, then the fully qualified SCAN would be similar to `docrac-scan.example.com`. You can use any name for the SCAN, if it is unique within your network and conforms to the RFC 952 standard.

8. If you configured the IP addresses in a DNS server, then, as the `root` user, change the hosts search order in `/etc/nsswitch.conf` on all nodes as shown here:

   Old:

   ```
   hosts:  files nis dns
   ```

   New:

   ```
   hosts:  dns files nis
   ```

9. After modifying the `nsswitch.conf` file, restart the `nscd` daemon on each node using the following command:

   ```
   # /sbin/service nscd restart
   ```

After you have completed the installation process, configure clients to use the SCAN to access the cluster. Using the previous example, the clients would use `docrac-scan` to connect to the cluster.

- Verifying the Network Configuration
  After you have configured the network, perform verification tests to make sure it is configured properly. If there are problems with the network connection between nodes in the cluster, then the Oracle Clusterware installation fails.

> **See Also:**
>
> - "About Network Hardware Requirements"
> - *Oracle Grid Infrastructure Installation and Upgrade Guide* for your platform

2.3.1 Verifying the Network Configuration

After you have configured the network, perform verification tests to make sure it is configured properly. If there are problems with the network connection between nodes in the cluster, then the Oracle Clusterware installation fails.

To verify the network configuration on a two-node cluster that is running Oracle Linux:

1. As the `root` user, verify the configuration of the public and private networks. Verify that the interfaces are configured on the same network (either private or public) on all nodes in your cluster.

   In this example, `eth0` is used for the public network and `eth1` is used for the private network on each node.

   ```
   # /sbin/ifconfig

   eth0      Link encap:Ethernet  HWaddr 00:0E:0C:08:67:A9
             inet addr: 192.0.2.100   Bcast:192.0.2.255   Mask:255.255.240.0
             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
             RX packets:270332689 errors:0 dropped:0 overruns:0 frame:0
             TX packets:112346591 errors:2 dropped:0 overruns:0 carrier:2
             collisions:202 txqueuelen:1000
             RX bytes:622032739 (593.2 MB)  TX bytes:2846589958 (2714.7 MB)
             Base address:0x2840 Memory:fe7e0000-fe800000

   eth1      Link encap:Ethernet  HWaddr 00:04:23:A6:CD:59
             inet addr: 10.10.10.11   Bcast: 10.10.10.255   Mask:255.255.240.0
             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
             RX packets:21567028 errors:0 dropped:0 overruns:0 frame:0
             TX packets:15259945 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:4091201649 (3901.6 MB)  TX bytes:377502797 (360.0 MB)
             Base address:0x2800 Memory:fe880000-fe8a0000
   ```

2. As the `root` user, verify the network configuration by using the `ping` command to test the connection from each node in your cluster to all the other nodes. For example, as the `root` user, you might run the following commands on each node:

   ```
   # ping -c 3 racnode1.example.com
   # ping -c 3 racnode1

   # ping -c 3 racnode2.example.com
   # ping -c 3 racnode2
   ```

   You should not get a response from the nodes using the `ping` command for the virtual IPs (`racnode1-vip`, `racnode2-vip`) or the SCAN IPs until after Oracle Clusterware is installed and running. If the `ping` commands for the public addresses fail, then resolve the issue before you proceed.

3. Ensure that you can access the default gateway with a `ping` command. To identify the default gateway, use the `route` command, as described in the Oracle Linux Help utility.

> **See Also:**
>
> - "Checking the Settings for the Interconnect"
> - "Configuring the Network"
> - "About Network Hardware Requirements"

2.4 Preparing the Operating System and Software

When you install the Oracle software on your server, Oracle Universal Installer expects the operating system to have specific packages and software applications installed.

- About Setting the Time on All Nodes
  Before starting the installation, ensure that the date and time settings on all the cluster nodes are set as closely as possible to the same date and time.
- About Configuring Kernel Parameters
  OUI checks the current settings for various kernel parameters to ensure they meet the minimum requirements for deploying Oracle RAC. For production database systems, Oracle recommends that you tune the settings to optimize the performance of your particular system.
- About Performing Platform-Specific Configuration Tasks
  You may be required to perform special configuration steps that are specific to the operating system on which you are installing Oracle RAC, or for the components used with your cluster.

> **See Also:**
>
> - "Preparing the Server"
> - "Verifying Operating System and Software Requirements"

2.4.1 About Setting the Time on All Nodes

Before starting the installation, ensure that the date and time settings on all the cluster nodes are set as closely as possible to the same date and time.

A cluster time synchronization mechanism ensures that the internal clocks of all the cluster members are synchronized. For Oracle RAC on Linux, you can use either the Network Time Protocol (NTP) or the Oracle Cluster Time Synchronization Service.

NTP is a protocol designed to synchronize the clocks of servers connected by a network. When using NTP, each server on the network runs client software to periodically make timing requests to one or more servers, referred to as reference NTP servers. The information returned by the timing request is used to adjust the server's clock. All the nodes in your cluster should use the same reference NTP server.

> **Note:**
>
> If you use NTP on Linux or UNIX platforms, then you must configure it using the `-x` flag. See the *Oracle Grid Infrastructure Installation and Upgrade Guide* for your platform for details on how to configure NTP.

If you do not configure NTP, then Oracle configures and uses the Cluster Time Synchronization Service (CTSS). CTSS can also be used to synchronize the internal clocks of all the members in the cluster. CTSS keeps the member nodes of the cluster synchronized. CTSS designates the first node in the cluster as the master and then synchronizes all other nodes in the cluster to have the same time as the master node. CTSS does not use any external clock for synchronization.

> **Note:**
>
> Using NTP or CTSS does not protect your system against human error resulting from a change in the system time for a node.

2.4.2 About Configuring Kernel Parameters

OUI checks the current settings for various kernel parameters to ensure they meet the minimum requirements for deploying Oracle RAC. For production database systems, Oracle recommends that you tune the settings to optimize the performance of your particular system.

> **Note:**
>
> If you find parameter settings or shell limit values on your system that are greater than the values mentioned by OUI, then *do not* modify the parameter setting.

2.4.3 About Performing Platform-Specific Configuration Tasks

You may be required to perform special configuration steps that are specific to the operating system on which you are installing Oracle RAC, or for the components used with your cluster.

The following list provides examples of operating-specific installation tasks:

- Configure the use of Huge Pages on SUSE Linux, Red Hat Enterprise Linux, or Oracle Linux.
- Set shell limits for the `oracle` user on Red Hat Linux or Oracle Linux systems to increase the number of files and processes available to Oracle Clusterware and Oracle RAC.
- Create X library symbolic links on HP-UX.

- Configure network tuning parameters on AIX Based Systems.

> **See Also:**
>
> - "Preparing the Server"
> - "Preparing the Operating System and Software"
> - "About Installing Oracle RAC on Different Operating Systems"
> - *Oracle Grid Infrastructure Installation Guide* for your platform

2.5 Configuring Installation Directories and Shared Storage

Oracle RAC requires access to a shared file system for storing Oracle Clusterware files. You must also determine where the Oracle software and database files will be installed.

You must complete certain storage configuration tasks before you start Oracle Universal Installer.

> **Note:**
>
> Additional configuration is required to use the Oracle Automatic Storage Management library Driver (ASMLIB) with third party vendor multipath disks. Refer to *Oracle Grid Infrastructure Installation Guide* for your operating system for more details about these requirements.

- About the Oracle Inventory Directory
  The Oracle Inventory (`oraInventory`) directory is the central inventory record of all Oracle software installations on a server.
- Locating the Oracle Inventory Directory
  If you have an existing Oracle Inventory, then ensure that you use the same Oracle Inventory for all Oracle software installations, and ensure that all Oracle software users you intend to use for installation have permissions to write to this directory.
- Creating the Oracle Grid Infrastructure for a Cluster Home Directory
  During installation, you are prompted to provide a path to a home directory in which to place the software for Oracle Grid Infrastructure for a cluster. OUI installs Oracle Clusterware and Oracle ASM into a directory referred to as *`Grid_home`*.
- Creating the Oracle Base Directory
  Before installing any Oracle software, you must configure an Oracle base directory. OUI uses the Oracle base directory to determine the location of the Oracle Inventory directory, and also for installing Oracle RAC.
- About the Oracle Home Directory
  The Oracle home directory is the location in which the Oracle RAC software is installed.

- Configuring Shared Storage
  Each node in a cluster requires external shared disks for storing the Oracle Clusterware (Oracle Cluster Registry and voting disk) files, and Oracle Database files.
- Creating Files on a NAS Device for Use with Oracle Automatic Storage Management
  If you have a certified NAS storage device, then you can create zero-padded files in an NFS mounted directory and use those files as disk devices in an Oracle ASM disk group.
- About Oracle ASM with Oracle ASM Filter Driver
  During Oracle Grid Infrastructure installation, you can choose to install and configure Oracle Automatic Storage Management Filter Driver (Oracle ASMFD). Oracle ASMFD helps prevent corruption in Oracle ASM disks and files within the disk group.
- Using Oracle ASMFD to Configure Disks for Oracle ASM
  Oracle ASM Filter Driver (Oracle ASMFD) simplifies the configuration and management of disk devices by eliminating the need to rebind disk devices used with Oracle ASM each time the system is restarted.

> **See Also:**
>
> - "About Operating System Users and Groups"
> - *Oracle Grid Infrastructure Installation and Upgrade Guide* for your platform

2.5.1 About the Oracle Inventory Directory

The Oracle Inventory (`oraInventory`) directory is the central inventory record of all Oracle software installations on a server.

The `oraInventory` directory contains the following:

- A registry of the Oracle home directories (Oracle Grid Infrastructure for a cluster and Oracle Database) on the system
- Installation logs and trace files from installations of Oracle software. These files are also copied to the respective Oracle homes for future reference

OUI creates the `oraInventory` directory during installation. By default, the Oracle Inventory directory is not installed under the Oracle Base directory. This is because all Oracle software installations share a common Oracle Inventory, so there is only one Oracle Inventory for all users, whereas each user that performs a software installation can use a separate Oracle base directory.

2.5.2 Locating the Oracle Inventory Directory

If you have an existing Oracle Inventory, then ensure that you use the same Oracle Inventory for all Oracle software installations, and ensure that all Oracle software users you intend to use for installation have permissions to write to this directory.

To determine if you have an Oracle central inventory directory (oraInventory) on your system:

1. Run the following command to check for an existing Oracle Inventory directory:

   ```
   # more /etc/oraInst.loc
   ```

   - If the `oraInst.loc` file exists, then the output from this command is similar to the following:

     ```
     inventory_loc=/u01/app/oraInventory
     inst_group=oinstall
     ```

     In the previous output example:

     – The `inventory_loc` group shows the location of the Oracle Inventory.

     – The `inst_group` parameter shows the name of the Oracle Inventory group (in this example, `oinstall`).

   - If the Oracle Inventory directory does not exist, then the command returns an error message indicating that the file or directory does not exist.

2. If the Oracle Inventory directory does not exist, then you do not have to create one before installing the Oracle software.

> **See Also:**
>
> "About the Oracle Inventory Directory"

2.5.3 Creating the Oracle Grid Infrastructure for a Cluster Home Directory

During installation, you are prompted to provide a path to a home directory in which to place the software for Oracle Grid Infrastructure for a cluster. OUI installs Oracle Clusterware and Oracle ASM into a directory referred to as *`Grid_home`*.

Ensure that the directory path you provide for the *`Grid_home`* meets the following requirements:

- It should be created in a path outside existing Oracle homes.
- It should not be located in a user home directory.
- It should be created either as a subdirectory in a path where all files can be owned by `root`, or in a unique path.

- If you create the path before installation, then it should be owned by the installation owner of Oracle Grid Infrastructure for a cluster (`oracle` or `grid`), and set to 775 permissions.

Before you start the installation, you must have sufficient disk space on a file system for the Oracle Grid Infrastructure for a cluster directory. The file system that you use for the Grid home directory must have at least 4.5 GB of available disk space.

The path to the Grid home directory must be the same on all nodes. As the `root` user, you should create a path compliant with Oracle Optimal Flexible Architecture (OFA) guidelines, so that OUI can select that directory during installation.

To create the Grid home directory:

Log in to the operating system as the `root` user, then enter the following commands, where *`grid`* is the name of the user that will install the Oracle Grid Infrastructure software:

```
# mkdir -p /u01/app/12.2.0/grid
# chown -R grid:oinstall /u01/app/12.2.0/grid
```

> **Note:**
>
> On Linux and UNIX systems, you must ensure the Grid home directory is not a subdirectory of the Oracle base directory. Installing Oracle Clusterware in an Oracle base directory causes installation errors.

Related Topics

- About Hardware Requirements
  Each node you make part of your cluster, or Oracle Clusterware and Oracle RAC installation, must satisfy the minimum hardware requirements of the software.
- Configuring Shared Storage
  Each node in a cluster requires external shared disks for storing the Oracle Clusterware (Oracle Cluster Registry and voting disk) files, and Oracle Database files.

2.5.4 Creating the Oracle Base Directory

Before installing any Oracle software, you must configure an Oracle base directory. OUI uses the Oracle base directory to determine the location of the Oracle Inventory directory, and also for installing Oracle RAC.

Oracle Universal Installer (OUI) creates the Oracle base directory for you in the location you specify. This directory is owned by the user performing the installation. The Oracle base directory (`ORACLE_BASE`) helps to facilitate the organization of Oracle installations, and helps to ensure that installations of multiple databases maintain an Optimal Flexible Architecture (OFA) configuration.

OFA guidelines recommend that you use a path similar to the following for the Oracle base directory:

```
/mount_point/app/user
```

In the preceding path example, the variable *mount_point* is the mount point directory for the file system where you intend to install the Oracle software and *user* is the Oracle software owner (typically oracle). For OUI to recognize the path as an Oracle software path, it must be in the form u0[1-9]/app, for example, /u01/app.

The path to the Oracle base directory must be the same on all nodes. The permissions on the Oracle base directory should be at least 750.

Assume you have determined that the file system mounted as /u01 has sufficient room for both the Oracle Grid Infrastructure for a cluster and Oracle RAC software. You have decided to make the /u01/app/oracle/ directory the Oracle base directory. The user installing all the Oracle software is the oracle user.

To create the Oracle base directory:

1. Log in as the root user.
2. Use the mkdir command to create the path to the Oracle base directory.

   ```
   # mkdir -p /u01/app/oracle/
   ```

3. Change the ownership of the Oracle base path to the Oracle software owner, oracle.

   ```
   # chown -R oracle:oinstall /u01/app/oracle/
   ```

4. Change the permissions on the Oracle base directory to 775.

   ```
   # chmod -R 775 /u01/app/oracle/
   ```

2.5.5 About the Oracle Home Directory

The Oracle home directory is the location in which the Oracle RAC software is installed.

You can use an Oracle home directory created in the local file system, for example, /u01/app/oracle/product/12.2.0/dbhome_1. The same directory must exist on every node in the cluster. You do not have to create these directories before installation. By default, the installer suggests a subdirectory of the Oracle base directory for the Oracle home.

You can also use a shared Oracle home. The location of the shared Oracle home can be on network storage, or a supported cluster file system such as Oracle Automatic Storage Management Cluster File System (Oracle ACFS). For more information about Oracle ACFS, see *Oracle Automatic Storage Management Administrator's Guide*.

If you use the local file system for the Oracle home directory, and you want to install a different release of Oracle RAC or Oracle Database on the same server, then you must use a separate Oracle home directory for each software installation. Multiple releases of the same product or different products can run from different Oracle homes concurrently. Products installed in one Oracle home do not conflict or interact with products installed in another Oracle home.

Using different Oracle homes for your installed software enables you to perform maintenance operations on the Oracle software in one home without effecting the software in another Oracle home. However, it also increases your software maintenance costs, because each Oracle home must be upgraded or patched separately.

2.5.6 Configuring Shared Storage

Each node in a cluster requires external shared disks for storing the Oracle Clusterware (Oracle Cluster Registry and voting disk) files, and Oracle Database files.

The supported types of shared storage depend upon the platform you are using, for example:

- Oracle Automatic Storage Management (strongly recommended)
- A supported cluster file system, such as OCFS2 for Linux, or General Parallel File System (GPFS) on IBM platforms
- Network file system (NFS), which is not supported on Linux on POWER or on IBM zSeries Based Linux
- (Upgrades only) Shared disk partitions consisting of block devices. Block devices are disk partitions that are not mounted using the Linux file system. Oracle Clusterware and Oracle RAC write to these partitions directly.

> **Note:**
>
> You cannot use OUI to install Oracle Clusterware files on block or raw devices. You cannot put Oracle Clusterware binaries and files on Oracle Automatic Storage Management Cluster File System (Oracle ACFS).

If you decide to use OCFS2 to store the Oracle Clusterware files, then you must use the proper version of OCFS2 for your operating system version. OCFS2 works with Oracle Linux and Red Hat Linux kernel version 2.6.

For all installations, you must choose the storage option to use for Oracle Clusterware files and Oracle Database files. The examples in this guide use Oracle ASM to store the Oracle Clusterware and Oracle Database files. The Oracle Grid Infrastructure for a cluster and Oracle RAC software is installed on disks local to each node, not on a shared file system.

This guide describes two different methods of configuring the shared disks for use with Oracle ASM:

- Creating Files on a NAS Device for Use with Oracle Automatic Storage Management
- Using Oracle ASMFD to Configure Disks for Oracle ASM

> **Note:**
>
> For the most up-to-date information about supported storage options for Oracle RAC installations, refer to the Certifications pages on My Oracle Support at:
>
> `https://support.oracle.com`

> **See Also:**
>
> - *Oracle Grid Infrastructure Installation and Upgrade Guide* for your platform if you are using a cluster file system or NFS

2.5.7 Creating Files on a NAS Device for Use with Oracle Automatic Storage Management

If you have a certified NAS storage device, then you can create zero-padded files in an NFS mounted directory and use those files as disk devices in an Oracle ASM disk group.

Ensure that you specify the ASM discovery path for Oracle ASM disks.

During installation of Oracle Grid Infrastructure 12*c* Release 2 (12.2), Oracle Universal Installer (OUI) can create files in the NFS mounted directory you specify. The following procedure explains how to manually create files in an NFS mounted directory to use as disk devices in an Oracle ASM disk group.

1. If necessary, create an exported directory for the disk group files on the NAS device.
2. Switch user to `root`.
3. Create a mount point directory on the local system.

   For example:

   ```
   # mkdir -p /mnt/oracleasm
   ```
4. To ensure that the NFS file system is mounted when the system restarts, add an entry for the file system in the mount file `/etc/fstab`.
5. Enter a command similar to the following to mount the NFS on the local system:

   ```
   # mount /mnt/oracleasm
   ```
6. Choose a name for the disk group to create, and create a directory for the files on the NFS file system, using the disk group name as the directory name.

   For example, if you want to set up a disk group for a sales database:

   ```
   # mkdir /mnt/oracleasm/sales1
   ```
7. Use commands similar to the following to create the required number of zero-padded files in this directory:

   ```
   # dd if=/dev/zero
   of=/mnt/oracleasm/sales1/disk1 bs=1024k
   count=1000
   ```

   This example creates 1 GB files on the NFS file system. You must create one, two, or three files respectively to create an external, normal, or high redundancy disk group.

> **Note:**
>
> Creating multiple zero-padded files on the same NAS device does not guard against NAS failure. Instead, create one file for each NAS device and mirror them using the Oracle ASM technology.

8. Enter commands similar to the following to change the owner, group, and permissions on the directory and files that you created:

   ```
   # chown -R grid:asmadmin /mnt/oracleasm
   # chmod -R 660 /mnt/oracleasm
   ```

   In this example, the installation owner is `grid` and the OSASM group is `asmadmin`.

9. During Oracle Database installations, edit the Oracle ASM disk discovery string to specify a regular expression that matches the file names you created.

   For example:

   ```
   /mnt/oracleasm/sales1/
   ```

Related Topics

- My Oracle Support Note 359515.1

2.5.8 About Oracle ASM with Oracle ASM Filter Driver

During Oracle Grid Infrastructure installation, you can choose to install and configure Oracle Automatic Storage Management Filter Driver (Oracle ASMFD). Oracle ASMFD helps prevent corruption in Oracle ASM disks and files within the disk group.

Oracle ASM Filter Driver (Oracle ASMFD) rejects write I/O requests that are not issued by Oracle software. This write filter helps to prevent users with administrative privileges from inadvertently overwriting Oracle ASM disks, thus preventing corruption in Oracle ASM disks and files within the disk group. For disk partitions, the area protected is the area on the disk managed by Oracle ASMFD, assuming the partition table is left untouched by the user.

Oracle ASMFD simplifies the configuration and management of disk devices by eliminating the need to rebind disk devices used with Oracle ASM each time the system is restarted.

If Oracle ASMLIB exists on your Linux system, then deinstall Oracle ASMLIB before installing Oracle Grid Infrastructure, so that you can choose to install and configure Oracle ASMFD during an Oracle Grid Infrastructure installation.

> **Note:**
>
> Oracle ASMFD is supported on Linux x86–64 and Oracle Solaris operating systems.

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*

2.5.9 Using Oracle ASMFD to Configure Disks for Oracle ASM

Oracle ASM Filter Driver (Oracle ASMFD) simplifies the configuration and management of disk devices by eliminating the need to rebind disk devices used with Oracle ASM each time the system is restarted.

The overall process of configuring disk with Oracle ASMFD is:

1. Unzip the Oracle Grid Infrastructure software into the Oracle Grid home directory.
2. Use the Oracle ASM command tool (ASMCMD) to label the disks to use for the disk group during installation.
3. Install and configure Oracle Grid Infrastructure. Select the disks and the option to use Oracle ASMFD during installation.

3

Installing Oracle Grid Infrastructure

Before installing Oracle Real Application Clusters (Oracle RAC) and Oracle RAC One Node using Oracle Universal Installer (OUI), you must first install the Oracle Grid Infrastructure for a cluster software, which consists of Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM).

- Using Rapid Home Provisioning to Install Oracle Grid Infrastructure
  Rapid Home Provisioning is a method of deploying software homes to any number of nodes in a data center from a single cluster, and also facilitates patching and upgrading software.
- Installing Oracle Grid Infrastructure for a Cluster
  The software for Oracle Grid Infrastructure for a cluster consists of Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM).
- About Verifying the Oracle Clusterware Installation
  Use Cluster Verification Utility (CVU) to verify that your installation is configured correctly.
- Confirming Oracle ASM Function for Oracle Clusterware Files
  Confirm Oracle ASM is running after installing Oracle Grid Infrastructure.

3.1 Using Rapid Home Provisioning to Install Oracle Grid Infrastructure

Rapid Home Provisioning is a method of deploying software homes to any number of nodes in a data center from a single cluster, and also facilitates patching and upgrading software.

With Rapid Home Provisioning, you create, store, and manage templates of Oracle homes as **images** (called gold images) of Oracle software, such as databases, middleware, and applications. You can make a **working copy** of any gold image, and then you can provision that working copy to any node in the data center.

You store the gold images in a repository located on a **Rapid Home Provisioning Server**, which runs on one server in the Rapid Home Provisioning Server cluster that is a highly available provisioning system. With a single command, Rapid Home Provisioning can provision new Grid homes to servers for Oracle Grid Infrastructure 11.2.0.4, 12.1.0.2 and 12.2. There are no prerequisites for the target servers. You do not need to install any client or agent software on the servers before the provisioning the Oracle Grid Infrastructure software.

> **See Also:**
>
> *Oracle Clusterware Administration and Deployment Guide* for more information about Rapid Home Provisioning

3.2 Installing Oracle Grid Infrastructure for a Cluster

The software for Oracle Grid Infrastructure for a cluster consists of Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM).

- About Cluster Verification Utility Fixup Scripts on Linux and UNIX
  During installation on Linux and UNIX platforms, for certain prerequisite check failures, you can click **Fix & Check Again** to generate a fixup script.
- Installing Oracle Grid Infrastructure for a New Cluster
  Complete this procedure to install Oracle Grid Infrastructure (Oracle Clusterware and Oracle Automatic Storage Management) on your cluster.
- Completing the Oracle Clusterware Configuration
  After you have installed Oracle Clusterware, verify that the node applications are started.

3.2.1 About Cluster Verification Utility Fixup Scripts on Linux and UNIX

During installation on Linux and UNIX platforms, for certain prerequisite check failures, you can click **Fix & Check Again** to generate a fixup script.

The installer detects when the minimum requirements for an installation are not met, and creates shell scripts, called Fixup scripts, to finish incomplete system configuration steps. If the installer detects an incomplete task, then it generates a Fixup script (`runfixup.sh`). You also can have CVU generate Fixup scripts before installation.

Fixup scripts do the following:

- If necessary, set kernel parameters to values required for successful installation, including:
  - Shared memory parameters.
  - Open file descriptor and UDP send/receive parameters.
- Create and set permissions on the Oracle Inventory (central inventory) directory.
- Create or reconfigure primary and secondary group memberships for the installation owner, if necessary, for the Oracle Inventory directory and the operating system privileges groups.
- Set shell limits if necessary to required values.

You can run the script after you click **Fix and Check Again**. You are prompted by the installer to run the fixup script as the `root` user in a separate session or you can specify through the installer interface that the scripts should be run automatically. You must run the script on all the nodes specified by the installer.

Modifying the contents of the generated fixup script is not recommended.

> **Note:**
>
> Using fixup scripts does not ensure that all the required prerequisites for installing Oracle Grid Infrastructure for a cluster and Oracle RAC are satisfied. You must still verify that all the requirements listed in Preparing Your Cluster are met to ensure a successful installation.

3.2.2 Installing Oracle Grid Infrastructure for a New Cluster

Complete this procedure to install Oracle Grid Infrastructure (Oracle Clusterware and Oracle Automatic Storage Management) on your cluster.

Starting with Oracle Grid Infrastructure 12*c* Release 2 (12.2), the installation media is replaced with a zip file for the Oracle Grid Infrastructure installer. Run the installation wizard after extracting the zip file into the target home path.

At any time during installation, if you have a question about what you are being asked to do, or what input you are required to provide during installation, click the **Help** button on the installer page.

You should have your network information, storage information, and operating system users and groups available to you before you start installation, and you should be prepared to run root scripts.

As the user that owns the software for Oracle Grid Infrastructure for a cluster (`grid`)on the first node, install Oracle Grid Infrastructure for a cluster. Note that the installer uses Secure Shell (SSH) to copy the binary files from this node to the other nodes during the installation. During installation, in the Cluster Node Information window, about you specify the nodes in your cluster, you can click **SSH Connectivity** and the installer configures SSH connectivity between the specified nodes for you.

> **Note:**
>
> These installation instructions assume you do not already have any Oracle software installed on your system. If you have already installed Oracle ASMLIB, then you will not be able to install Oracle ASM Filter Driver (Oracle ASMFD) until you uninstall Oracle ASMLIB. You can use Oracle ASMLIB instead of Oracle ASMFD for managing the disks used by Oracle ASM, but those instructions are not included in this guide.

To install the software for Oracle Grid Infrastructure for a cluster:

1. As the `grid` user, download the Oracle Grid Infrastructure image files and unzip the files into the Grid home. For example:

   ```
   mkdir -p /u01/app/12.2.0/grid
   chown grid:oinstall /u01/app/12.2.0/grid
   cd /u01/app/12.2.0/grid
   unzip -q download_location/grid_home_image.zip
   ```

> **Note:**
>
> - You must extract the zip image software into the directory where you want your Grid home to be located.
> - Download and copy the Oracle Grid Infrastructure image files to the local node only. During installation, the software is copied and installed on all other nodes in the cluster.

2. Configure the shared disks for use with Oracle ASM Filter Driver:
   1. Log in as the root user and set the environment variable `$ORACLE_HOME` to the location of the Grid home and the environment variable `$ORACLE_BASE` to a temporary location.

      ```
      su root
      set ORACLE_HOME=/u01/app/12.2.0/grid
      set ORACLE_BASE=/tmp
      ```

      You set `$ORACLE_BASE` to a temporary location to avoid creating diagnostic or trace files in the Grid home before the Oracle Grid Infrastructure installation.
   2. Use Oracle ASM command line tool (ASMCMD) to provision the disk devices for use with Oracle ASM Filter Driver.

      ```
      ./u01/app/12.2.0/grid/bin/asmcmd afd_label DATA1 /dev/sdb --init
      ./u01/app/12.2.0/grid/bin/asmcmd afd_label DATA1 /dev/sdc --init
      ./u01/app/12.2.0/grid/bin/asmcmd afd_label DATA1 /dev/sdd --init
      ```
   3. Verify the device has been marked for use with Oracle ASMFD.

      ```
      ./u01/app/12.2.0/grid/bin/asmcmd afd_lslbl /dev/sdb
      ./u01/app/12.2.0/grid/bin/asmcmd afd_lslbl /dev/sdc
      ./u01/app/12.2.0/grid/bin/asmcmd afd_lslbl /dev/sdd
      ```
   4. Unset the ORACLE_BASE environment variable.

      ```
      unset ORACLE_BASE
      ```
3. Start the Oracle Grid Infrastructure wizard by running the following command:

   ```
   Grid_home/gridSetup.sh
   ```

   The installer starts and the Select Configuration Option window appears.
4. Choose the option **Configure Grid Infrastructure for a New Cluster**, then click **Next**.

   The Select Cluster Type window appears.
5. Choose the option **Configure an Oracle Standalone Cluster** , then click **Next**.

   The Grid Plug and Play Information window appears.
6. In the **Cluster Name** and **SCAN Name** fields, enter the names for your cluster and cluster scan that are unique throughout your entire enterprise network, then click **Next**.

   For example, you might choose a name that is based on the node names' common prefix. This guide uses the cluster name `docrac` and the cluster SCAN name of `docrac-scan`.

   The Cluster Node Information screen appears.

7. In the Public Hostname column of the table of cluster nodes, you should see your local node, for example `racnode1.example.com`.

   a. Click **Add** to add another node to the cluster.

   b. Enter the second node's public name (`racnode2`), and virtual IP name (`racnode2-vip`), then click **OK**.

      You are returned to the Cluster Node Information window. You should now see both nodes listed in the table of cluster nodes. Make sure the Role column is set to HUB for both nodes.

   c. Make sure both nodes are selected, then click the **SSH Connectivity** button at the bottom of the window.

      The bottom panel of the window displays the SSH Connectivity information.

   d. Enter the operating system user name and password for the Oracle software owner (`grid`). Select the option **If you have configured SSH connectivity between the nodes**, then select the **Reuse private and public keys existing in user home** option. Click **Setup**.

      A message window appears, indicating that it might take several minutes to configure SSH connectivity between the nodes. After a short period, another message window appears indicating that passwordless SSH connectivity has been established between the cluster nodes. Click **OK** to continue.

   e. When returned to the Cluster Node Information window, click **Next** to continue.

   The Specify Network Interface Usage page appears.

8. Select the usage type for each network interface displayed, then click **Next**.

   Verify that each interface has the correct interface type associated with it. If you have network interfaces that should not be used by Oracle Clusterware, then set the network interface type to **Do Not Use**. For example, if you have only two network interfaces, then set the public interface to have a Use For value of **Public** and set the private network interface to have a Use For value of **ASM & Private**.

   Make sure both nodes are selected, then click **Next**. The Grid Infrastructure Management Repository Option window appears.

9. Choose whether you want to store the Grid Infrastructure Management Repository in a separate Oracle ASM disk group, then click **Next**.

   The Storage Option Information window appears.

10. Select the option **Use Oracle Flex ASM for storage**, then click **Next** to continue.

    The Grid Infrastructure Management Repository Option window appears.

11. Select the **Local** option, then click **Next** to continue

    The Create ASM Disk Group window appears.

12. Provide the name and specifications for the Oracle ASM disk group.

    a. In the **Disk Group Name** field, enter a name for the disk group, for example DATA.

    b. Choose the Redundancy level for this disk group. Normal is the recommended option.

    c. In the Add Disks section, choose the disks to add to this disk group.

In the Add Disks section you should see the disks that you labeled in Step 2. If you do not see the disks, click the **Change Discovery Path** button and provide a path and pattern match for the disk, for example, `/dev/sd*`

d. Check the option **Configure Oracle ASM Filter Driver**.

When you have finished providing the information for the disk group, click **Next**.

The Specify ASM Password window appears.

13. Choose to use the same password for the Oracle ASM SYS and ASMSNMP account, or specify different passwords for each account, then click **Next**.

    The Failure Isolation Support window appears.

14. Select the option **Do not use Intelligent Platform Management Interface (IMPI)**, then click **Next**.

    The Specify Management Options window appears.

15. If you have Enterprise Manager Cloud Control installed in your enterprise, then choose the option **Register with Enterprise Manager (EM) Cloud Control** and provide the EM configuration information. If you do not have Enterprise Manager Cloud Control installed in your enterprise, then click **Next** to continue.

    The Privileged Operating System Groups window appears.

16. Accept the default operating system group names for Oracle ASM administration and click **Next**.

    The Specify Install Location window appears.

17. Specify the directory to use for the Oracle base for the Oracle Grid Infrastructure installation, then click **Next**.

    If you copied the Oracle Grid Infrastructure installation files into the Oracle Grid home directory as directed in Step 1, then the default location for the Oracle base directory should display as `/u01/app/grid`.

    If you have not installed Oracle software previously on this computer, then the Create Inventory window appears.

18. Change the path for the inventory directory, if required. Then, click **Next**.

    If you are using the same directory names as the examples in this book, then it should show a value of `/u01/app/oraInventory`. The group name for the `oraInventory` directory should show `oinstall`.

    The Root Script Execution Configuration window appears.

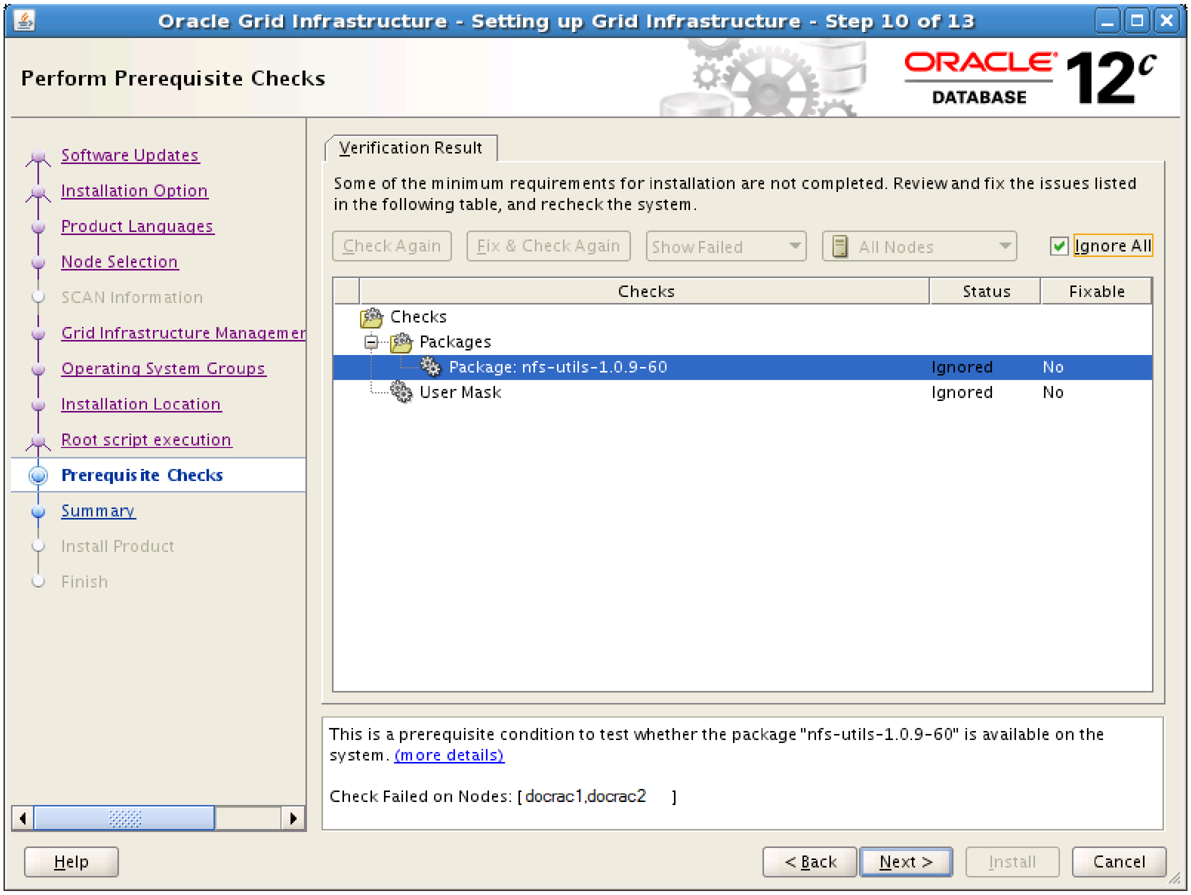
19. Select the option to **Automatically run configuration scripts.** Enter the credentials for the root user or a sudo account, then click **Next**.

    Alternatively, you can Run the scripts manually as the `root` user at the end of the installation process when prompted by the installer.

    The Perform Prerequisite Checks window appears.

20. If any of the checks have a status of Failed and are not Fixable, then you must manually correct these issues. After you have fixed the issue, you can click the **Check Again** button to have the installer recheck the requirement and update the status. Repeat as needed until all the checks have a status of Succeeded. Click **Next**.

Figure 3-1 Perform Prerequisite Checks Window



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

Checks	Status	Fixable
Checks		
Packages		
Package: nfs-utils-1.0.9-60	Ignored	No
User Mask	Ignored	No

This is a prerequisite condition to test whether the package "nfs-utils-1.0.9-60" is available on the system. (more details)

Check Failed on Nodes: [docrac1,docrac2]

The Summary window appears.

21. Review the contents of the Summary window and then click **Install**.

    The installer displays a progress indicator enabling you to monitor the installation process.

22. If you did not configure automation of the root scripts, then you are required to run certain scripts as the `root` user, as specified in the Execute Configuration Scripts window appears. Do *not* click **OK** until you have run the scripts. Run the scripts on all nodes as directed, in the order shown.

    For example, on Oracle Linux you perform the following steps (note that for clarity, the examples show the current user, node and directory in the prompt):

    a. As the `oracle` user on `racnode1`, open a terminal window, and enter the following commands:

    ```
    [oracle@racnode1 oracle]$ cd /u01/app/oraInventory
    [oracle@racnode1 oraInventory]$ su
    ```

    b. Enter the password for the `root` user, and then enter the following command to run the first script on `racnode1`:

    ```
    [root@racnode1 oraInventory]# ./orainstRoot.sh
    ```

c. After the `orainstRoot.sh` script finishes on `racnode1`, open another terminal window, and as the `oracle` user, enter the following commands:

```
[oracle@racnode1 oracle]$ ssh racnode2
[oracle@racnode2 oracle]$ cd /u01/app/oraInventory
[oracle@racnode2 oraInventory]$ su
```

d. Enter the password for the `root` user, and then enter the following command to run the first script on `racnode2`:

```
[root@racnode2 oraInventory]# ./orainstRoot.sh
```

e. After the `orainstRoot.sh` script finishes on `racnode2`, go to the terminal window you opened in part a of this step. As the `root` user on `racnode1`, enter the following commands to run the second script, `root.sh`:

```
[root@racnode1 oraInventory]# cd /u01/app/12.2.0/grid
[root@racnode1 grid]# ./root.sh
```

Press **Enter** at the prompt to accept the default value.

> **Note:**
>
> You must run the `root.sh` script on the first node and wait for it to finish. You can run `root.sh` scripts concurrently on all other nodes except for the last node on which you run the script. Like the first node, the `root.sh` script on the last node must be run separately.

f. After the `root.sh` script finishes on `racnode1`, go to the terminal window you opened in part c of this step. As the `root` user on `racnode2`, enter the following commands:

```
[root@racnode2 oraInventory]# cd /u01/app/12.2.0/grid
[root@racnode2 grid]# ./root.sh
```

After the `root.sh` script completes, return to the OUI window where the Installer prompted you to run the `orainstRoot.sh` and `root.sh` scripts. Click **OK**.

The software installation monitoring window reappears.

23. Continue monitoring the installation until the Finish window appears. Then click **Close** to complete the installation process and exit the installer.

> **Caution:**
>
> After installation is complete, do not remove manually or run `cron` jobs that remove `/tmp/.oracle` or `/var/tmp/.oracle` directories or their files while Oracle software is running on the server. If you remove these files, then the Oracle software can encounter intermittent hangs. Oracle Clusterware installations can fail with the error:
>
> ```
> CRS-0184: Cannot communicate with the CRS daemon.
> ```

3.2.3 Completing the Oracle Clusterware Configuration

After you have installed Oracle Clusterware, verify that the node applications are started.

Depending on which operating system you use, you may have to perform some postinstallation tasks to configure the Oracle Clusterware components properly.

To complete the Oracle Clusterware configuration on Oracle Linux:

1. As the `grid` user on the first node, check the status of the Oracle Clusterware targets by entering the following command:

   ```
   /u01/app/12.2.0/grid/bin/crsctl check cluster -all
   ```

   This command provides output showing if all the required cluster services, such as `gsd`, `ons`, and `vip`, are started on the nodes of your cluster.

2. In the displayed output, you should see the Oracle Clusterware daemons are online for each node in the cluster.

   ```
   ******************************************************************
   racnode1:
   CRS-4537: Cluster Ready Services is online
   CRS-4529: Cluster Synchronization Services is online
   CRS-4533: Event Manager is online
   ******************************************************************
   racnode2:
   CRS-4537: Cluster Ready Services is online
   CRS-4529: Cluster Synchronization Services is online
   CRS-4533: Event Manager is online
   ******************************************************************
   ```

   If you see that one or more Oracle Clusterware resources are offline, or are missing, then the Oracle Clusterware software did not install properly.

> **Note:**
>
> Avoid changing host names after you complete the Oracle Clusterware installation, including adding or deleting domain qualifications. Nodes with changed host names must be deleted from the cluster and added back with the new name.

3.3 About Verifying the Oracle Clusterware Installation

Use Cluster Verification Utility (CVU) to verify that your installation is configured correctly.

After the Oracle Clusterware installation is complete, OUI automatically runs Cluster Verification Utility (CVU) as a configuration assistant to verify that the Oracle Clusterware installation has been completed successfully.

If CVU reports problems with your configuration, then correct these errors before proceeding.

> **See Also:**
>
> *Oracle Clusterware Administration and Deployment Guide* for more information about using CVU and resolving configuration problems

3.4 Confirming Oracle ASM Function for Oracle Clusterware Files

Confirm Oracle ASM is running after installing Oracle Grid Infrastructure.

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

```
srvctl status asm
```

For example:

```
srvctl status asm
ASM is running on node1,node2, node3, node4
```

> **Note:**
>
> 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.

4

Installing the Oracle Database Software and Creating a Database

After Oracle Grid Infrastructure for a cluster is functional, you can install the Oracle Database software on the nodes of your cluster.

Oracle Universal Installer (OUI) installs the software on the local node and then copies the binary files from the local node to all the other nodes in the cluster.

- Using Rapid Home Provisioning to Install Oracle Database Software
  Rapid Home Provisioning is a method of deploying software homes to any number of nodes in a data center from a single cluster, and also facilitates patching and upgrading software.
- Using Oracle Enterprise Manager Cloud Control for Provisioning Operations
  As part of the database provisioning solution, Oracle Enterprise Manager Cloud Control enables you to provision Oracle Databases (also known as single-instance databases) and Oracle Real Application Clusters (Oracle RAC) databases.
- Verifying My Oracle Support Credentials
  When you first start OUI, you are prompted to enter your e-mail address and My Oracle Support password.
- Configuring the Operating System Environment
  You run Oracle Universal Installer (OUI) from the `oracle` user account. You don’t have to configure the environment of the `oracle` user before you start OUI.
- Creating Additional Oracle ASM Disk Groups
  If you want to use a separate disk group for your Oracle database files or for the fast recovery area, then you must create the additional Oracle ASM disk groups before installing Oracle Database software.
- Deciding Between Multitenant Container Databases and Non-CDBs in Oracle RAC
  Review the information to decide how to deploy your Oracle Database using the CDB or Non-CDB options.
- Using Oracle Universal Installer to Install Oracle RAC Software
  After you have configured the operating system environment, you can use Oracle Universal Installer to install the Oracle RAC software and create an Oracle RAC database.
- Verifying Your Oracle RAC Database Installation
  If you chose to create an Oracle RAC database during installation, then verify that all the database services are up and running.
- Performing Postinstallation Tasks
  After you have installed the Oracle Real Application Clusters (Oracle RAC) software and created an Oracle RAC database there are additional tasks to perform before your cluster database is ready for use. These steps are recommended, but are not required.

- Creating an Oracle RAC Database Using DBCA
  The Database Configuration Assistant (DBCA) utility is used to create Oracle Databases.

> **See Also:**
>
> - "Installing Oracle Grid Infrastructure for a Cluster"

4.1 Using Rapid Home Provisioning to Install Oracle Database Software

Rapid Home Provisioning is a method of deploying software homes to any number of nodes in a data center from a single cluster, and also facilitates patching and upgrading software.

With Rapid Home Provisioning, you create, store, and manage templates of Oracle homes as **images** (called gold images) of Oracle software, such as databases, middleware, and applications. You can make a **working copy** of any gold image, and then you can provision that working copy to any node in the data center.

You store the gold images in a repository located on a **Rapid Home Provisioning Server**, which runs on one server in the Rapid Home Provisioning Server cluster that is a highly available provisioning system. With a single command, Rapid Home Provisioning can provision new Oracle Database homes for the various releases of Oracle Database, such as 11.2.0.x (Oracle Database 11*g*), 12.1.0.2 and 12.2.0.1 (Oracle Database 12*c*). There are no prerequisites for the target servers. You do not need to install any client or agent software on the servers before the provisioning the Oracle Grid Infrastructure software.

When you provision such software, Rapid Home Provisioning offers additional features for creating various types of databases (such as Oracle Real Application Clusters (Oracle RAC), single instance, and Oracle Real Application Clusters One Node (Oracle RAC One Node) databases) on different types of storage, and other options, such as using templates and creating container databases (CDBs).

> **See Also:**
>
> *Oracle Clusterware Administration and Deployment Guide* for more information about Rapid Home Provisioning

4.2 Using Oracle Enterprise Manager Cloud Control for Provisioning Operations

As part of the database provisioning solution, Oracle Enterprise Manager Cloud Control enables you to provision Oracle Databases (also known as single-instance databases) and Oracle Real Application Clusters (Oracle RAC) databases.

Using Oracle Enterprise Manager Cloud Control, you can extend or delete Oracle RAC nodes, and provision Oracle RAC One node databases. Oracle Enterprise Manager Cloud Control also helps in setting up the environment for provisioning and does various pre-checks on the host to verify provisioning is successful. Before you can use Oracle Enterprise Manager Cloud Control to provision your Oracle software, you must perform the following system configuration tasks:

- Discover all host by installing the Enterprise Manager Agent on all nodes.
- Configure named and preferred credentials in Enterprise Manager.
- Configure a software library and setup roles.
- Download software from Oracle Technology Network to create installation media components in Enterprise Manager. Optionally, you can create a Provisioning Profile (similar to a gold image) from an existing system.

Details on how to perform these tasks, and how to provision your Oracle Grid Infrastructure and Oracle RAC software using Oracle Enterprise Manager Cloud Control are available in *Oracle Enterprise Manager Lifecycle Management Administrator's Guide*.

4.3 Verifying My Oracle Support Credentials

When you first start OUI, you are prompted to enter your e-mail address and My Oracle Support password.

By entering this information, you enable the following features:

- Oracle Configuration Manager is installed and configured. This option enables you to associate information about your Oracle RAC configuration with your My Oracle Support account. In the event that you must place a service request with Oracle Support, that configuration information can help provide a more rapid resolution to the service issue.
- You can receive e-mail notification of security alerts from My Oracle Support.
- Automatic download and application of the most recent patches to the newly installed Oracle software (with Oracle Grid Infrastructure for a cluster or Oracle Database). The software updates that can be downloaded include patches, critical patch updates, installer updates, and patch sets.

If you choose to enable these features, then you must supply your My Oracle Support account name (your e-mail address) and your password. You may have to configure the proxy settings before your computer can connect to My Oracle Support.

If you have downloaded the software updates, then during installation you can enter the directory location where the files are stored on the local server instead of downloading them from My Oracle Support. The software updates are applied to the installed software during the installation process.

4.4 Configuring the Operating System Environment

You run Oracle Universal Installer (OUI) from the `oracle` user account. You don’t have to configure the environment of the `oracle` user before you start OUI.

You can set the `ORACLE_BASE` environment variable to the directory in which you want the Oracle Inventory files for the Oracle Database software located. For example, if you plan to make the Oracle Database home directory `/u01/app/oracle/`

product/12.2.0/dbhome_1, then you would set ORACLE_BASE to the directory /u01/app/oracle/. If you set the ORACLE_BASE environment variable before installation, then the path specified becomes the default location for the central inventory displayed by OUI.

You can also set the ORACLE_HOME environment variable to the location chosen for the Oracle Database home. If you set the ORACLE_HOME environment variable before installation, then the path specified becomes the default location for the Oracle home displayed by OUI.

(Optional) To modify the user environment before installing Oracle Database software on Oracle Linux:

1. As the oracle user, execute the following commands:

   ```
   [oracle]$ unset ORACLE_SID
   [oracle]$ export ORACLE_BASE=/u01/app/oracle/
   [oracle]$ export ORACLE_HOME=/u01/app/oracle/product/12.2.0/dbhome_1
   ```

2. Verify the changes have been made by executing the following commands:

   ```
   [oracle]$ echo $ORACLE_SID

   [oracle]$ echo $ORACLE_HOME
   /u01/app/oracle/product/12.2.0/dbhome_1
   [oracle]$ echo $ORACLE_BASE
   /u01/app/oracle/
   ```

4.5 Creating Additional Oracle ASM Disk Groups

If you want to use a separate disk group for your Oracle database files or for the fast recovery area, then you must create the additional Oracle ASM disk groups before installing Oracle Database software.

If you chose to store the Oracle Clusterware files on Oracle ASM during the Oracle Grid Infrastructure for a cluster installation, then a single disk group was created in Oracle ASM. You can use this same disk group to store the data files for your Oracle database, or you can create a separate disk group for the Oracle Database files.

To create an additional disk group using ASMCA:

1. Prepare the disks or devices for use with Oracle ASM, as described in "Configuring Installation Directories and Shared Storage ".
2. Start the Oracle Automatic Storage Configuration Assistant (ASMCA) from the Grid home:

   ```
   /u01/app/12.2.0/grid/bin/asmca
   ```

   ASMCA starts, and displays the Disk Groups window.
3. Click the **Create** button to create a disk group.

   The Create Disk Group window appears.
4. Provide the following information:
   - In the **Disk Group Name** field, enter a name for the new disk group, for example, FRA, for a disk group that stores the fast recovery files.
   - Choose a **Redundancy** level, for example, Normal.

- Select the disks to include in the new disk group.

After you have provided all the information, click **OK**. A progress window titled DiskGroup: Creation appears. After a few minutes, a message appears indicating the disk group was created successfully. Click **OK** to continue.

5. Repeat Step 3 and 4 to create additional disk groups, or click **Exit**, then select **Yes** to exit the utility.

> **See Also:**
>
> - "About Oracle Automatic Storage Management"
> - "Creating Files on a NAS Device for Use with Oracle Automatic Storage Management"
> - *Oracle Grid Infrastructure Installation and Upgrade Guide for Linux*

4.6 Deciding Between Multitenant Container Databases and Non-CDBs in Oracle RAC

Review the information to decide how to deploy your Oracle Database using the CDB or Non-CDB options.

Starting in Oracle Database 12*c* Release 1 (12.1), you must create a database as either a multitenant container database (CDB) or as an Oracle database that is non-CDB. This also applies to Oracle RAC databases. The only difference to the installation process is to choose whether to create the Oracle RAC database as a CDB or non-CDB.

A pluggable database (PDB) is a portable collection of schemas, schema objects, and nonschema objects that appears to an Oracle Net client as a non-CDB. PDBs can be plugged into CDBs. A CDB can contain multiple PDBs. Each PDB appears on the network as a separate database.

If you create an Oracle RAC database as a CDB and plug one or more PDBs into the CDB, then, by default, a PDB is not started automatically on any instance of the Oracle RAC CDB. With the first dynamic database service assigned to the PDB (other than the default database service which has the same name as the database name), the PDB is made available on those instances on which the service runs.

Whether or not a PDB is available on more than one instance of an Oracle RAC, CDB is typically managed by the services running on the PDB. You can manually enable PDB access on each instance of an Oracle RAC CDB by starting the PDB manually on that instance.

See Also:

- *Oracle Database Concepts* for more information about PDB concepts
- *Oracle Database Administrator's Guide* for more information about managing PDBs
- *Oracle Real Application Clusters Administration and Deployment Guide* for information specific to the administration of Oracle RAC CDBs

4.7 Using Oracle Universal Installer to Install Oracle RAC Software

After you have configured the operating system environment, you can use Oracle Universal Installer to install the Oracle RAC software and create an Oracle RAC database.

To install Oracle Database software on your cluster and to create a database:

1. As the `oracle` user, use the following commands to start OUI, where *`staging_area`* is the location of the staging area on disk, or the location of the mounted installation disk:

   ```
   cd /staging_area
   ./runInstaller
   ```

   The Configure Security Updates window appears.

2. Optional: Enter your email address and My Oracle Support password, then click **Next** to continue.

   If you want to receive notification by email of any newly discovered security issues related to the software you are installing, then enter an email address in the **Email** field. If you also want to receive the security updates through My Oracle Support, then use the same email address that is registered with My Oracle Support, select the **I wish to receive security updates via My Oracle Support** option, and enter your My Oracle Support login password in the **My Oracle Support Password** field.

   If you provide an email address, then the Oracle Configuration Manager (OCM) tool will also be installed. This utility provides Oracle Support Services with configuration information about your system when creating service requests. You can disable the OCM tool after installation, but Oracle strongly discourages this. OCM does not access, collect, or store any personal information (except for technical support contact information), or any business data files residing in your software environment. For more information about OCM, see `http://www.oracle.com/technetwork/documentation/ocm-092152.html`.

   After you click **Next**, the Select Installation Option window appears.

3. Select **Create and configure a database**, then click **Next** to continue:

   The System Class window appears.

4. Choose **Server Class**, then click **Next**.

If you choose the Desktop Class option, then OUI installs a single-instance database, not a clustered database.

The Grid Installation Options window appears.

5. Choose either the **Oracle Real Application Clusters database installation** option or the **Oracle RAC One Node database installation** option.

   The Select Install Type window appears.

6. Choose **Admin-managed** or **Policy-Managed** for your database, then click **Next** to continue.

   If you chose Admin-Managed, then the Select List of Nodes window appears. If you selected Policy-Managed, then the Select Install Type window appears.

7. For an administrator-managed database, you must select the nodes in the cluster where you want to run Oracle RAC or Oracle RAC One Node.

   a. Select the nodes on which you want to install Oracle Database software and create an Oracle RAC instance. All the available nodes in the cluster are selected by default.

      > **Note:**
      >
      > Select both nodes during installation, even if you are creating an Oracle RAC One Node database.

   b. Click the **SSH Connectivity** button at the bottom of the window.

      The bottom panel of the window displays the SSH Connectivity information.

8. Because you configured SSH connectivity between the nodes for the Oracle Grid Infrastructure for a cluster installation, select the **Reuse private and public keys existing in user home option**. Click **Test**.

   If you are using a network user that has a home directory on shared storage, then also select the **User home if shared by the selected nodes** option.

   A message window appears, indicating that passwordless SSH connectivity has been established between the cluster nodes. Click **OK** to continue.

9. When returned to the Node Selection window, click **Next** to continue.

   The Select Install Type window appears.

10. Choose the **Typical install** option, and click **Next**.

    A typical installation requires minimal input. It installs the software and optionally creates a general-purpose database. If you choose the Advanced installation type (not documented in this guide), then you are prompted to provide more information about how the database should be configured. For example, you could set passwords for each user account individually, choose a different template for the starter database, choose a nondefault language for the database, and so on.

    The Typical Install Configuration window appears.

11. In this window, you must provide the following information:

    - **Oracle base location**: The default value is `/u01/app/oracle/`. If you did not set the `ORACLE_BASE` environment variable and the default location is different from the directory location you have chosen, then enter the directory for your Oracle base or click the Browse button to change the directory path.

- **Software location**: If you did not set the ORACLE_HOME environment variable before starting the installation, then enter the directory for your Oracle home or click the Browse button to change the directory path.
- **Storage Type**: In this drop-down list, choose Automatic Storage Management (ASM). If you do not want to use Oracle ASM, then choose File System. Because Oracle ASM was installed with the Oracle Grid Infrastructure for a cluster, Oracle Automatic Storage Management is the default value.
- **Database file location**: Choose the disk group to use for storing the database files. You can use the same disk group that Oracle Clusterware uses. If you do not want to use the same disk group that is currently being used to store the Oracle Clusterware files, then you must exit the installation and create a new disk group using Oracle ASM utilities. Refer to "Creating Additional Oracle ASM Disk Groups" for more information on creating a disk group.

  If you chose the File System storage type, then enter the directory location of the shared storage where the database files will be created.
- **ASMSNMP Password**: Enter the password for the ASMSNMP user. The ASMSNMP user is used primarily by Oracle Enterprise Manager to monitor Oracle ASM instances. See *Oracle Automatic Storage Management Administrator's Guide* for more information about the ASMSNMP user.

  If you chose File System for the Storage Type, then this field is disabled.
- **Database edition**: From this drop-down list choose either Enterprise Edition or Standard Edition. The number in parentheses next to your choice indicates the amount of disk space required.
- **OSDBA Group**: From this drop-down list select the operating system group used for database administration, for example, dba.
- **Global database name**: Enter the fully qualified global name for your database. The global database name is in the form *DB_UNIQUE_NAME.DB_DOMAIN*, for example, orcl.example.com.

  > **Note:**
  >
  > The first eight characters of the ORACLE_SID must be a unique identifier for each database. The SID prefix cannot contain underscore (_), dollar ($), or pound (#) characters. The total length of the ORACLE_SID cannot exceed 15 characters for Oracle RAC databases.

- **Administrative password**: Enter the password to be used for the administrative account, such as SYS, SYSTEM, and DBSNMP.
- **Confirm Password**: Enter the same password in this field.

12. After you have provided all the necessary information, click **Next**.

    The Perform Prerequisite Checks window appears.

13. After a short period of time, the Summary window appears. Review the information on this window, then click **Finish** to continue.

If any of the information in the Summary window is incorrect, then use the Back button to return to a previous window and correct it.

After you click Finish, OUI displays a progress indicator to show that the installation has begun. This step takes several minutes to complete.

14. After the software is installed on each node, if you select the option to create a database, then OUI starts the Database Configuration Assistant (DBCA).

    This utility creates the database using the global database name specified in Step 9. At the end of the database creation, you see the DBCA window with the database configuration information displayed.

    There is also a Password Management button that you can click to unlock the database user accounts, or change their default passwords.

    After making note of the information in the Summary window, click **OK**.

    OUI configures Oracle Configuration Management, if you provided the information in Step 2.

    If you chose to perform a software-only installation, then the database configuration assistants are not started. You must run DBCA separately to create the Oracle RAC One Node database.

    > **See Also:**
    >
    > *Oracle Real Application Clusters Installation Guide* for your platform for information about creating an Oracle RAC One Node database using DBCA

15. Perform the following steps to run the `root.sh` script (note that for clarity, the examples show the current user, node and directory in the prompt):

    In the last step of the installation process, you are prompted to perform the task of running the `root.sh` script on both nodes, as specified in the Execute Configuration Scripts window. Do *not* click **OK** until you have run the scripts on all nodes.

    a. Open a terminal window as the `oracle` user on the first node. Change directories to your Oracle home directory, and then switch to the `root` user by entering the following commands:

    ```
    [oracle@racnode1 oracle]$ cd /u01/app/oracle/product/12.2.0/dbhome_1
    [oracle@racnode1 dbhome_1]$ su
    ```

    b. Enter the password for the `root` user, and then run the script specified in the Execute Configuration scripts window:

    ```
    [root@racnode1 dbhome_1]# ./root.sh
    ```

    > **Note:**
    >
    > You can run the `root.sh` script simultaneously on all nodes in the cluster for Oracle RAC installations or upgrades.

    c. After responding to prompt, press the **Enter** key. To accept the default choice, press the Enter key without entering any text.

As the root.sh script runs, it prompts you for the path to the local bin directory. The information displayed in the brackets is the information it has obtained from your system configuration. It also writes the dbhome, oraenv, and coraenv files in the /usr/local/bin directory. If these files exist, then you are asked to overwrite them.

d. Enter commands similar to the following to run the script on the other nodes:

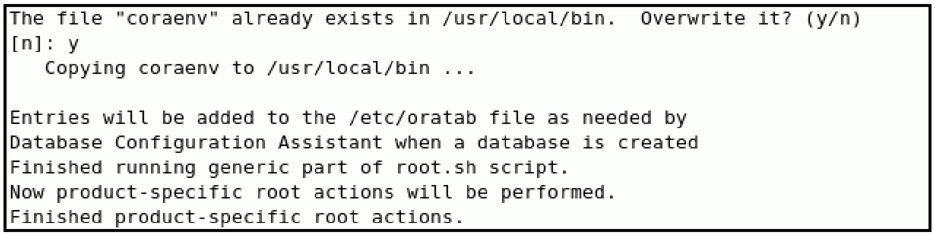
```
[root@racnode1 dhome_1]# exit
[oracle@racnode1 dhome_1]$ ssh racnode2
[oracle@racnode2 ~]$ cd /u01/app/oracle/product/12.2.0/dbhome_1
[oracle@racnode2 dbhome_1]$ su
```

e. Enter the password for the root user, and then run the script specified in the Execute Configuration scripts window:

```
[root@racnode2 dbhome_1]# ./root.sh
```

f. After responding to each prompt, press the **Enter** key.

g. When the root.sh script completes, the following messages are displayed:



```
The file "coraenv" already exists in /usr/local/bin.  Overwrite it? (y/n)
[n]: y
   Copying coraenv to /usr/local/bin ...

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

16. After you finish executing the script on all nodes, return to the Execute Configuration scripts window and click **OK**.

    The Install Product window displays.

17. Click **Next** to complete the installation.

    The Finish window is displayed.

18. Click **Close** to exit the installer.

4.8 Verifying Your Oracle RAC Database Installation

If you chose to create an Oracle RAC database during installation, then verify that all the database services are up and running.

To verify the Oracle RAC database services are started:

1. Log in as the oracle user and go to the *Grid_home*/bin directory:

   ```
   [oracle] $ cd /u01/app/12.2.0/grid/bin
   ```

2. Run the following command to view the status of the resources managed by Oracle Clusterware that contain the string 'ora':

   ```
   [oracle] $ ./crsctl status resource -w "TYPE co 'ora'" -t
   ```

   The output of the command should show that the Oracle Clusterware, Oracle ASM, and Oracle Database resources are available (online) for each host. An example of the output is:

```
---------------------------------------------------------------------------
NAME              TARGET STATE    SERVER            STATE_DETAILS
---------------------------------------------------------------------------
Local Resources
---------------------------------------------------------------------------
ora.DATA.dg
                  ONLINE  ONLINE   racnode1
                  ONLINE  ONLINE   racnode2

ora.LISTENER.lsnr
                  ONLINE  ONLINE   racnode1
                  ONLINE  ONLINE   racnode2

ora.asm

                  ONLINE  ONLINE   racnode1          Started
                  ONLINE  ONLINE   racnode2          Started

ora.eons
                  ONLINE  ONLINE   racnode1
                  ONLINE  ONLINE   racnode2

ora.gsd
                  OFFLINE OFFLINE   racnode1
                  OFFLINE OFFLINE   racnode2

ora.net1.network
                  ONLINE  ONLINE   racnode1
                  ONLINE  ONLINE   racnode2

ora.ons
                  ONLINE  ONLINE   racnode1
                  ONLINE  ONLINE   racnode2
ora.registry.acfs
                  ONLINE  ONLINE   racnode1
                  ONLINE  ONLINE   racnode2
---------------------------------------------------------------------------
Cluster Resources
---------------------------------------------------------------------------
ora.LISTENER_SCAN1.lsnr
      1           ONLINE  ONLINE   racnode1
ora.oc4j
      1           OFFLINE OFFLINE
ora.orcl.db
      1           ONLINE  ONLINE   racnode1            Open
      2           ONLINE  ONLINE   racnode2            Open
ora.racnode1.vip
      1           ONLINE  ONLINE   racnode1
ora.racnode2.vip
      1           ONLINE  ONLINE   racnode2
ora.scan1.vip
      1           ONLINE  ONLINE   racnode1
```

Caution:

After installation is complete, do not remove manually or run `cron` jobs that remove `/tmp/.oracle` or `/var/tmp/.oracle` directories or their files while Oracle software is running on the server. If you remove these files, then the Oracle software can encounter intermittent hangs. Oracle Clusterware installations will fail with the error:

```
CRS-0184: Cannot communicate with the CRS daemon.
```

See Also:

"Performing Postinstallation Tasks"

4.9 Performing Postinstallation Tasks

After you have installed the Oracle Real Application Clusters (Oracle RAC) software and created an Oracle RAC database there are additional tasks to perform before your cluster database is ready for use. These steps are recommended, but are not required.

- About Downloading and Installing Oracle Database Patches
  Ensure that you run the latest patch set of the installed software.
- Recompiling All Invalid Objects
  Oracle recommends you run the `utlrp.sql` script after you install, patch, or upgrade a database, to identify and recompile invalid objects.
- About Configuring User Accounts
  You can use different operating system accounts for accessing and managing your Oracle RAC database.
- Configuring Services on Oracle RAC and Oracle RAC One Node CDBs
  During installation, if you select a multitenant container database (CDB), and configure pluggable databases (PDBs), then you must add services to the PDBs after installation.
- Downloading and Installing the ORAchk Health Check Tool
  Download and install the ORAchk utility to perform proactive heath checks for the Oracle software stack.

See Also:

- "Verifying Your Oracle RAC Database Installation"

4.9.1 About Downloading and Installing Oracle Database Patches

Ensure that you run the latest patch set of the installed software.

Periodically, Oracle issues bug fixes for its software called **patches**. **Patch sets** are a collection of bug fixes that were produced up to the time of the patch set release. Patch sets are fully tested product fixes. Application of a patch set affects the software residing in your Oracle home.

If you configured access to My Oracle Support during installation, then the latest patches should have been downloaded and applied during installation.

If you did not configure access to My Oracle Support within OUI, then you should apply the latest patch set for your release and any necessary patches that are not included in a patch set. Information about downloading and installing patches and patch sets is covered in Managing Oracle Software and Applying Patches.

> **See Also:**
>
> *Oracle Grid Infrastructure Installation Guide for Linux* for detailed instructions on how to locate and download patches and patch sets

4.9.2 Recompiling All Invalid Objects

Oracle recommends you run the `utlrp.sql` script after you install, patch, or upgrade a database, to identify and recompile invalid objects.

The `utlrp.sql` script recompiles all invalid objects. Run the script immediately after installation, to ensure that users do not encounter invalid objects.

1. Start SQL*Plus:

```
sqlplus "/ AS SYSDBA"
```

2. Run the `utlrp.sql` script, where *`Oracle_home`* is the Oracle home path:

```
SQL> @Oracle_home/rdbms/admin/utlrp.sql
```

The `utlrp.sql` script automatically recompiles invalid objects in either serial or parallel recompilation, based on both the number invalid objects, and on the number of CPUs available. CPUs are calculated using the number of CPUs (`cpu_count`) multiplied by the number of threads for each CPU (`parallel_threads_per_cpu`). On Oracle Real Application Clusters (Oracle RAC), this number is added across all Oracle RAC nodes.

4.9.3 About Configuring User Accounts

You can use different operating system accounts for accessing and managing your Oracle RAC database.

The `oracle` user operating system account is the account that you used to install the Oracle Database software. You can modify the shell configuration file to set environment variables such as ORACLE_HOME whenever you log in as that operating system user.

See Also:

About Operating System Users and Groups

4.9.4 Configuring Services on Oracle RAC and Oracle RAC One Node CDBs

During installation, if you select a multitenant container database (CDB), and configure pluggable databases (PDBs), then you must add services to the PDBs after installation.

If you do not add services to PDBs, and then the Oracle RAC One Node CDB fails over to another node, or you manually relocate the CDB to another node, then by default, all PDBs associated with the CDB that do not have registered services are restarted in MOUNTED state.

PDBs are opened in Read-Write mode after failover, in case of Oracle RAC One Node database, or relocation in case of any Oracle RAC database, only after you have configured the PDBs to have associated services. If you have not associated services to PDBs, then the PDBs remains in MOUNTED state when the CDB instance restarts.

To add services to a PDB, use the following srvctl command syntax, where *cdbname* is the name of the CDB, *service_name* is the name of the service, and *pdbname* is the name of the PDB:

```
srvctl add service -d cdbname -s service_name -pdb pdbname
```

After you add services to your PDBs, if you relocate the CDB with which the PDBs are associated, or the CDB fails over for Oracle RAC One Node databases, then the PDBs associated with that CDB automatically open in Read-Write state.

See Also:

Oracle Database Concepts and *Oracle Real Application Clusters Administration and Deployment Guide* for more information about PDBs and adding services

4.9.5 Downloading and Installing the ORAchk Health Check Tool

Download and install the ORAchk utility to perform proactive heath checks for the Oracle software stack.

ORAchk replaces the RACCheck utility. ORAchk extends health check coverage to the entire Oracle software stack, and identifies and addresses top issues reported by Oracle users. ORAchk proactively scans for known problems with Oracle products and deployments, including the following:

- Standalone Oracle Database
- Oracle Grid Infrastructure
- Oracle Real Application Clusters
- Maximum Availability Architecture (MAA) Validation
- Upgrade Readiness Validations
- Oracle Golden Gate

Oracle is continuing to expand checks, based on customer requests.

ORAchk is supported on Windows Server 2012 and Windows Server 2016 on a Cygwin environment only.

Oracle recommends that you download and run the latest version of ORAchk from My Oracle Support. For information about downloading, configuring and running ORAchk utility, refer to My Oracle Support note 1268927.2:

https://support.oracle.com/epmos/faces/DocContentDisplay?id=1268927.2&parent=DOCUMENTATION&sourceId=USERGUIDE

Related Topics

- *Oracle ORAchk and EXAchk User's Guide*

4.10 Creating an Oracle RAC Database Using DBCA

The Database Configuration Assistant (DBCA) utility is used to create Oracle Databases.

If you did not select the option to create an Oracle Real Application Clusters (Oracle RAC) database during installation, then you will need to create one using DBCA after you have verified that the installation of the Oracle RAC software was successful. The steps for creating an Oracle RAC database are documented in *Oracle Real Application Clusters Installation Guide for Linux and UNIX*.

5

Converting an Existing Database to Oracle RAC or Oracle RAC One Node

You can convert a single-instance database to an Oracle RAC database or an Oracle Real Application Clusters (Oracle RAC) One Node database. You can also convert an Oracle Real Application Clusters (Oracle RAC) One Node database to an Oracle RAC database.

- About Converting an Oracle Database to an Oracle RAC Database
  If you have a single-instance database and you want to convert to Oracle RAC or Oracle RAC One Node, then you can use Oracle Enterprise Manager or Database Configuration Assistant (DBCA) to assist you with the task of converting a single-instance database installation to an Oracle Real Application Clusters (Oracle RAC) database.
- Converting a Single Instance Database to Oracle RAC Using DBCA
  If you have a single instance database running on a cluster node, you can convert the database to Oracle RAC or Oracle RAC One Node using Database Configuration Assistant (DBCA).
- Converting an Oracle RAC Database into an Oracle RAC One Node Database
  After you convert a single-instance Oracle database into a single-node Oracle RAC database, you can use the `srvctl` utility to convert the database into an Oracle RAC One Node database.
- Converting a Database from Oracle RAC One Node to Oracle RAC
  You can convert an Oracle RAC One Node database to an Oracle RAC database.

5.1 About Converting an Oracle Database to an Oracle RAC Database

If you have a single-instance database and you want to convert to Oracle RAC or Oracle RAC One Node, then you can use Oracle Enterprise Manager or Database Configuration Assistant (DBCA) to assist you with the task of converting a single-instance database installation to an Oracle Real Application Clusters (Oracle RAC) database.

The Convert to Cluster Database option in Oracle Enterprise Manager Cloud Control provides a GUI conversion tool. Additionally, after you have converted your single-instance database to an Oracle RAC database, you can use the `srvctl` utility to convert the database to an Oracle RAC One Node database.

- About Preparing for Database Conversion
  Before you start the process of converting your database to a cluster database, your database environment must meet certain prerequisites.
- Overview of the Database Conversion Process Using Cloud Control
  You can convert a single-instance database to an Oracle RAC database using Oracle Enterprise Manager Cloud Control.

5.1.1 About Preparing for Database Conversion

Before you start the process of converting your database to a cluster database, your database environment must meet certain prerequisites.

- The existing database and the target Oracle RAC database must be on the same release of Oracle Database 12*c* and must be running on the same platform.
- The hardware and operating system software used to implement your Oracle RAC database must be certified for use with the release of the Oracle RAC software you are installing.
- You must configure shared storage for your Oracle RAC database. You will need additional shared storage for the Oracle Clusterware files, such as the Oracle Cluster Registry, the voting files, and the Grid Infrastructure Management Repository.
- You must verify that any applications that run against the Oracle RAC database do not need any additional configuration before they can be used successfully with the cluster database. This applies to both Oracle applications and database features, such as Oracle Streams, and applications and products that do not come from Oracle.
- Backup procedures should be available before converting from a single-instance Oracle Database to Oracle RAC.
- For archiving in Oracle RAC environments, the archive log file format requires a thread number.
- The archived redo log files from all instances of an Oracle RAC database are required for media recovery. If you archive to a file and you do not use a cluster file system, or some other means to provide shared file systems, then you require a method of accessing the archived redo log files from all nodes on which the cluster database has instances.

> **Note:**
>
> For information about using individual Oracle Database 12*c* database products or options, refer to the product documentation library, which is available on the Oracle Help Center website at `http://docs.oracle.com/en/database/`

> **See Also:**
>
> - "About Installing Oracle RAC on Different Operating Systems"
> - "Verifying System Requirements"
> - "Configuring Installation Directories and Shared Storage "

5.1.2 Overview of the Database Conversion Process Using Cloud Control

You can convert a single-instance database to an Oracle RAC database using Oracle Enterprise Manager Cloud Control.

This section summarizes the process of converting a single-instance database to an Oracle RAC database:

- Complete the prerequisite tasks for converting to an Oracle RAC database:
  - Oracle Clusterware and Oracle Database software is installed on all target nodes.
  - Oracle Clusterware is started.
  - The Oracle Database binary is enabled for Oracle RAC on all target nodes.
  - Shared storage is configured and accessible from all nodes.
  - User equivalency is configured for the operating system user performing the conversion.
  - Enterprise Manager agents are configured and running on all nodes, and are configured with the cluster and host information.
  - The database being converted has been backed up successfully.
- Access the Database Home page for the database you want to convert. For instructions on navigating to the Database Home page, see "Accessing the Database Home Page" in the *Oracle Database 2 Day DBA*Guide.
- From the **Availability** menu, select **Convert to Cluster Database**.
- Provide the necessary credentials.
- Select the host nodes that should contain instances of the new database.
- Provide listener and instance configuration information.
- Specify the location of the shared storage to be used for the data files.
- Submit the job.
- Complete the post-conversion tasks.

The resulting Oracle RAC database uses a server pool instead of a fixed configuration.

> **See Also:**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for a complete description of this process

5.2 Converting a Single Instance Database to Oracle RAC Using DBCA

If you have a single instance database running on a cluster node, you can convert the database to Oracle RAC or Oracle RAC One Node using Database Configuration Assistant (DBCA).

1. Change directory to `$ORACLE_HOME/bin`.
2. Start DBCA.

   ```
   $ dbca
   ```

3. In the Welcome window, click **Next**.
4. On the Operations page, select **Manage Templates**, and click **Next**
5. On the Template Management page, select **Create a database template** and **From an existing database (structure as well as data)**, then click **Next**
6. On the Source Database page, select the database name in the Database instance list, and click **Next**
7. In a separate window, use SQL to ensure that all pluggable databases (PDBs) are open, as follows:

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

   If any of the PDBs are in a state other than OPEN, then open them using SQL.
8. On the Template Properties page, enter a name for your template in the Name field. Oracle recommends that you use the database name.

   By default, the template files are generated in the directory `$ORACLE_HOME/assistants/dbca/templates`. You can enter a description of the file in the Description field. Change the template file location in the Template data file field to a temporary location outside of the Oracle home.

   When you have completed the entries, click **Next**.
9. On the Location of Database Related Files page, select **Maintain the file locations**, so that you can restore the database to the current directory structure, and click **Finish**.

   DBCA generates two files: a database structure file (`*template_name*.dbc`), and a database preconfigured image file (`*template_name*.dfb`).
10. Start DBCA again.
11. From the Welcome window, select **Oracle RAC One Node database**.
12. On the DBCA Template Selection page, use the template that you copied to a temporary location in Step 8. Use the **Browse** option to select the template location.
13. Select the option that you want to deploy.

    Your choices are the following: Oracle RAC database; Oracle RAC One Node database; or Oracle single-instance database.

5.3 Converting an Oracle RAC Database into an Oracle RAC One Node Database

After you convert a single-instance Oracle database into a single-node Oracle RAC database, you can use the `srvctl` utility to convert the database into an Oracle RAC One Node database.

To convert your database to an Oracle RAC One Node database, use the following command:

```
srvctl convert database -db database_name -dbtype RACONENODE
```

An Oracle RAC One Node database must be part of a multi-node cluster to support failover or online database relocation. You must either install Oracle Grid Infrastructure for a cluster and Oracle RAC on at least two nodes, or add a node to your existing single-node Oracle RAC database.

See Also:

- "Adding and Deleting Nodes and Instances " for more information about adding nodes
- "About Oracle RAC One Node" for more information about Oracle RAC One Node

5.4 Converting a Database from Oracle RAC One Node to Oracle RAC

You can convert an Oracle RAC One Node database to an Oracle RAC database.

You can perform the conversion by logging in as the Oracle RAC One Node database owner and entering the following SRVCTL command:

```
srvctl convert database -db db_unique_name -dbtype RAC
```

If you are relocating the database you want to convert to Oracle RAC using online database relocation, or an online database relocation has failed, then you must either quit or complete the relocation before you run the `srvctl convert database` command.

Converting an administrator-managed Oracle RAC One Node database to an Oracle RAC database configures all database services so that the single-instance database is the preferred instance for that service. After you convert the database, you can add instances to your database by using the `srvctl add instance` command.

Converting a policy-managed Oracle RAC One Node database to an Oracle RAC database sets all database services to UNIFORM cardinality. It also results in reusing the server pool in which the database currently runs. The conversion reconfigures the database to run on all nodes in the server pool. The command does not start any additional instances but running the `srvctl start database` command starts the database on all nodes in the server pool.

> **See Also:**
>
> `srvctl convert database` for more information about this command

6 Administering Database Instances and Cluster Databases

Web-based Oracle Enterprise Manager Cloud Control enables you to manage Oracle Real Application Clusters (Oracle RAC) databases. You can use the Cloud Control console to administer multiple Oracle RAC databases and cluster nodes.

The Enterprise Manager console is a central point of control for the Oracle environment. Using this console you can perform many tasks, such as starting and stopping database components and managing parameters and parameter files in Oracle RAC

- About Oracle Real Application Clusters Database Management
  There are differences when managing an Oracle RAC database compared to a single-instance Oracle Database.
- About Oracle RAC Management Using Enterprise Manager
  The web-based Oracle Enterprise Manager Cloud Control console enables you to manage Oracle RAC and Oracle RAC One Node databases.
- About Oracle RAC One Node Database Management
  Oracle RAC One Node databases are administered slightly differently from Oracle RAC or single-instance databases.
- About Password Files for an Oracle RAC or Oracle RAC One Node Database
  The database uses a password file to keep track of database user names that have been granted the SYSDBA, SYSOPER, SYSRAC, SYSBACKUP, SYSDG, or SYSKM administrative privilege.
- Starting and Stopping Oracle RAC Databases and Database Instances
  You can start up and shut down the cluster database from the Enterprise Manager Cluster Database Home page.
- Starting and Stopping PDBs in Oracle RAC
  You manage pluggable databases (PDBs) in an Oracle RAC-based multitenant database by managing services, regardless of whether the PDBs are policy-managed or administrator-managed.
- Managing Oracle Real Application Clusters Initialization Parameters
  Managing initialization parameters for an Oracle RAC database is essentially the same as managing them for a single-instance Oracle database.
- Limiting CPU Usage of Oracle RAC Database Instances
  Oracle Database provides a method for managing CPU allocations on a multi-CPU server running multiple database instances. This method is called instance caging.
- About Administering Storage in Oracle RAC
  This topic provides additional information for using Enterprise Manager to manage some storage structures of an Oracle RAC database.

See Also:

Oracle Database 2 Day DBA for an overview of database monitoring tasks

6.1 About Oracle Real Application Clusters Database Management

There are differences when managing an Oracle RAC database compared to a single-instance Oracle Database.

Oracle Real Application Clusters (Oracle RAC) provides technology that links two or more individual computers so that they function as one system. Oracle RAC enables each **node**, which is a computer that is a member of a cluster, to share access to the Oracle database. If one cluster node fails or is taken offline, then the other cluster nodes continue operating and the entire Oracle RAC database remains available. Two or more inexpensive computers can appear to applications as if they were a single, much more powerful and more expensive, computer.

To increase the performance of an Oracle RAC database, you can add cluster nodes. Each additional node can help speed up application processing, support more users or processes, or both. In addition, you can also add cluster nodes to increase the availability and reliability of a two-node Oracle RAC database. The more nodes that your Oracle RAC environment has, the less the impact of the loss of any individual node on the database.

An Oracle RAC database requires three components: cluster nodes, shared storage, and Oracle Clusterware. Although you can choose how many nodes your cluster should have and what type of shared storage to use, this guide describes one specific configuration for a two-node cluster. This two-node configuration uses Oracle Automatic Storage Management (Oracle ASM) for storage management and Recovery Manager (RMAN) for the backup and recovery strategy.

Most administration tasks are the same for Oracle single-instance and Oracle RAC databases. This guide provides additional instructions for database administration tasks specific to Oracle RAC, and recommendations for managing Oracle RAC databases.

Note:

If you are using Oracle Database Standard Edition, then your cluster must adhere to the license restrictions. See *Oracle Database Licensing Information* for specific details on these restrictions.

See Also:

Oracle Database 2 Day DBA for information about managing single instance databases

6.2 About Oracle RAC Management Using Enterprise Manager

The web-based Oracle Enterprise Manager Cloud Control console enables you to manage Oracle RAC and Oracle RAC One Node databases.

Enterprise Manager is a central point of control for the Oracle environment that you access by way of a graphical user interface (GUI). You can use Enterprise Manager to create and modify services, and to start and stop the cluster database instances and the cluster database. Use Enterprise Manager Cloud Control to administer your entire Oracle RAC environment, not just the Oracle RAC database.

When you log in to Oracle Enterprise Manager Cloud Control and access the database home page for an Oracle RAC database, the Cluster Database Home page appears. The Cluster Database Home page is similar to a single-instance Database Home page. However, on the Cluster Database Home page, Enterprise Manager displays the system state and availability of the entire Oracle RAC environment. This includes a summary about alert messages and job activity, and the status of and links to all the database and Oracle Automatic Storage Management (Oracle ASM) instances. By clicking the Cluster tab on this page you can view the Cluster Home page to view the status of and alerts for the underlying cluster.

Note:

Oracle Enterprise Manager Cloud Control manages all the Oracle ASM instances on the nodes. Only one Oracle ASM instance exists on a node, no matter how many Oracle databases use Oracle ASM for storage.

6.3 About Oracle RAC One Node Database Management

Oracle RAC One Node databases are administered slightly differently from Oracle RAC or single-instance databases.

Oracle Real Application Clusters One Node (Oracle RAC One Node) is a single instance of an Oracle Real Application Clusters (Oracle RAC) database that runs on one node in a cluster. Instead of stopping and starting instances, you use Oracle RAC One Node online database relocation to relocate the Oracle RAC One Node instance to another server.

- Administrator-managed Oracle RAC One Node databases: You must monitor the candidate node list and make sure a server is always available for failover if possible. Candidate servers reside in the Generic server pool of the cluster. If the node currently hosting the Oracle RAC One node database instance fails, then the database and its services fail over to one of the candidate servers.

- Policy-managed Oracle RAC One Node databases: You must ensure that the server pools are configured such that a server will be available for the database to fail over to in case its current node becomes unavailable. Also, the destination node for online database relocation must be located in the database's server pool.

For an overview of the online database relocation procedure, see the following topic:

- Online Database Relocation
  You can relocate an Oracle RAC One Node database to another node while still maintaining service availability using the online database relocation feature.

> **See Also:**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for more information about managing Oracle RAC One Node databases

6.3.1 Online Database Relocation

You can relocate an Oracle RAC One Node database to another node while still maintaining service availability using the online database relocation feature.

Only during a planned online database relocation is a second instance of an Oracle RAC One Node database created, so that any database sessions can continue while the database is relocated to a new node. You can only use online database relocation with Oracle RAC One Node databases but you cannot use online database relocation with Oracle RAC databases regardless of their management style (either administrator or policy managed).

You can use the `srvctl relocate database` command to relocate the Oracle RAC One Node database instance to a different node in the cluster. You can also use this command to configure the amount of time after the relocated database starts and services are migrated, before the former instance of the database stops. This configured amount of time is not an upper bound on the amount of time taken by the entire operation, but only controls how long the relocated database waits for connections to migrate from the former instance to the new instance, before stopping the former instance.

If your Oracle RAC One Node database is administrator managed, then the target node to which you want to relocate the database instance must be part of the Free server pool at the moment you initiate the relocation. If you have a spare target node in the Free server pool, then this node is added to the candidate list for the Oracle RAC One Node database.

Before you relocate a database instance, you must ensure that the database service user is added to the wallet. Run a command similar to the following to check whether the database service user is in the wallet:

```
crsctl query wallet -type OSUSER -all
```

If the database service user is not in the wallet, then run a command similar to the following to add the database service user to the wallet:

```
crsctl add wallet -type OSUSER -user user_name -passwd
```

- Use the `srvctl relocate database` command to initiate relocation of an Oracle RAC One Node database. For example:

  ```
  srvctl relocate database -db racdb -node node2
  ```

Online database relocation occurs, as follows:

1. Start a new database instance in a different location.
2. Move all the services to the relocated instance.
3. Wait for all the connections to migrate to the relocated instance.
4. Shut down the former database instance, forcing any remaining connections to move to the relocated instance.

The online relocation time-out, specified with the `-timeout` option of the `srvctl relocate database` command, is the amount of time you configure to complete step 3.

> **Note:**
>
> When you use password file-based authentication for remote management of Oracle RAC One Node databases without any shared password file, you must have two password files on each node where the database can run: one named `*SID_prefix*_1` and the other named `*SID_prefix*_2`. You must recopy both of these files to all candidate nodes every time you update the password file. This is true for both policy-managed and administrator-managed databases.
>
> Oracle recommends using Oracle Clusterware to start and stop the database, and defining users in the data dictionary for other management.

6.4 About Password Files for an Oracle RAC or Oracle RAC One Node Database

The database uses a password file to keep track of database user names that have been granted the SYSDBA, SYSOPER, SYSRAC, SYSBACKUP, SYSDG, or SYSKM administrative privilege.

Because these privileges allow access to a database instance even when the database is not open, control of these privileges is totally outside of the database itself. Users can be authenticated through operating system (OS) authentication, password file, or with a directory-based authentication service, such as Oracle Internet Directory.

Note:

Operating system authentication takes precedence over password file authentication. If you meet the requirements for operating system authentication, then even if you use a password file, you will be authenticated by operating system authentication.

If you are not in the one of the operating system groups, and you are not in the password file, then attempting to connect with the clause fails.

You can use password file authentication for an Oracle database instance and for an Oracle Automatic Storage Management (Oracle ASM) instance. The password file for an Oracle database is called a database password file, and the password file for Oracle ASM is called an Oracle ASM password file.

Starting with Oracle Database 12*c* release 1 the password file is stored in an Oracle ASM diskgroup. If you use DBCA or the installer to create the Oracle RAC or Oracle RAC One Node database, then, by default, the password file is created in an ASM diskgroup. This allows the password file to be accessible from any node in the cluster. If you create a password file manually, then it must be available to all the Oracle RAC database instances.

See Also:

Oracle Database Administrator’s Guide

6.5 Starting and Stopping Oracle RAC Databases and Database Instances

You can start up and shut down the cluster database from the Enterprise Manager Cluster Database Home page.

By using the Enterprise Manager Cluster Database Home page for cluster database startup and shutdown operations, you ensure that all the instances that belong to the Oracle RAC database are in a consistent state. Oracle Enterprise Manager enables you to more easily manage an Oracle RAC database.

You can start and stop individual instances in an Oracle RAC database. Starting and stopping one instance in an Oracle RAC database does not stop or start the other database instances. To completely stop an Oracle RAC database, you must shut down all of its instances.

You can also start and shut down instances using SQL*Plus or Server Control (SRVCTL).

To start and stop an entire Oracle RAC database, assuming you are using a server parameter file (SPFILE):

1. In Oracle Enterprise Manager, go to the Cluster Database Home page, logging in as user SYS.

See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. In the Cluster Database Home page, in the General section, click **Startup** if the database is down, or **Shutdown** if the database is started.

   The Startup/Shutdown: Specify Credentials page appears.

3. Enter the host credentials for the cluster nodes. The host credentials are the user name and password for a user who is a member of the `OSDBA` or `OSOPER` operating system group.

   The Startup/Shutdown: Select Operation page appears.

4. Click **Select All** to select all the instances, or then click **Shutdown** to stop all the database instances or **Startup** to start all the database instances.

   To start and stop individual instances, on the Startup/Shutdown: Select Operation page, select the database instances, then click **Startup** or **Shutdown** to perform the desired operation on the selected database instances. You can also start and shut down instances using SQL*Plus or Server Control (SRVCTL).

   The Startup/Shutdown: Confirmation page appears.

5. Click **Yes**.

> **Note:**
>
> You can start and shut down individual instances from each instance's home page. However, it is easier to perform instance startup and shutdown operations directly from the Startup/Shutdown: Select Operation page.

> **See Also:**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for more information about using command-line interfaces to start and stop Oracle RAC database instances

6.6 Starting and Stopping PDBs in Oracle RAC

You manage pluggable databases (PDBs) in an Oracle RAC-based multitenant database by managing services, regardless of whether the PDBs are policy-managed or administrator-managed.

Administering an Oracle RAC-based multitenant container database (CDB) is somewhat similar to administering a non-CDB. The differences are only that some administrative tasks apply to the entire CDB, some apply only to the root, and some apply to specific pluggable databases (PDBs). Administering a pluggable database (PDB) involves a small subset of the tasks required to administer a non-CDB. In this subset of tasks, most are the same for a PDB and a non-CDB. There are some differences, however, such as when you modify the open mode of a PDB. Also, a PDB administrator is limited to managing a single PDB and is not affected by other PDBs in the CDB.

You manage PDBs in an Oracle RAC-based CDB by managing services, regardless of whether the PDBs are policy managed or administrator managed. Assign one dynamic database service to each PDB to coordinate start, stop, and placement of PDBs across instances in a clustered container database.

For example, if you have a CDB called `raccont` with a policy-managed PDB called `spark`, which is in a server pool called `prod`, then assign a service called `plugsvc` to this database using the following command:

```
srvctl add service –db raccont –pdb spark –service plug –serverpool prod
```

The service `plugsvc` will be uniformly managed across all nodes in the server pool. If you want to have this service running as a singleton service in the same server pool, use the `-cardinality` singleton parameter with the preceding command.

- To open the PDB, you must start the service assigned to the PDB, plug, as follows:

  ```
  srvctl start service -db db_name -service service_name
  ```

  So, for the PDB named `spark`, and its service named `plugsvc`, you would run the following command:

  ```
  srvctl start service -db raccont -service plugsvc
  ```

- To stop the service, and restrict access to the PDB, run the following command:

  ```
  srvctl stop service -db db_name -service service_name
  ```

  So, for the PDB named `spark`, and its service named `plugsvc`, you would run the following command:

  ```
  srvctl stop service -db raccont -service plugsvc
  ```

  The PDB remains open until you close the PDB.

- To close the PDB, use the `ALTER PLUGGABLE DATABASE` command:

  ```
  ALTER PLUGGABLE DATABASE pdb_name CLOSE IMMEDIATE;
  ```

  If the service associated with the PDB is in the `ONLINE` state when Oracle Clusterware is shut down on a server hosting this service, then the service will be restored to its original state after the restart of Oracle Clusterware on this server. This way, starting PDBs is automated as with any other Oracle RAC database.

> **Note:**
>
> Unlike SQL*Plus, SRVCTL operates on a cluster database, as a whole. Starting a PDB using services therefore applies to multiple instances of the clustered CDB at the same time, if the service is defined to run on multiple servers simultaneously, and the current status of the cluster allows for this placement.

6.7 Managing Oracle Real Application Clusters Initialization Parameters

Managing initialization parameters for an Oracle RAC database is essentially the same as managing them for a single-instance Oracle database.

Note the following differences for parameters in Oracle RAC databases:

- Parameters that are cluster-specific have the value Cluster Database in the Category column.
- Parameters that are the same on each instance in the Oracle RAC database are identified in the Instance column with an asterisk (*).
- Parameters that are set to different values on each instance of an Oracle RAC database are listed by instance name.

One difference when managing initialization parameters in an Oracle RAC environment is that parameters can apply to more than one instance of the database. If you change a parameter setting that is marked by an asterisk, which indicates that the parameter is a cluster-wide database initialization parameter, then you change that parameter's setting for all the instances in your Oracle RAC database. If you change an initialization parameter prefixed with an instance name, or an instance-specific initialization parameter, then the change affects only that instance; the change does not affect the parameter's settings on other database instances.

This section contains the following topics:

- About Configuring Initialization Parameters for an Oracle RAC Database
  Oracle Database stores initialization parameters in either a server parameter file or a text-based parameter file.
- About Modifying the SERVICE_NAMES Parameter for Oracle RAC
  The `SERVICE_NAMES` parameter should not be modified in the same manner as other initialization parameters.
- About the Server Parameter File for Oracle Real Application Clusters
  A server parameter file is a repository for initialization parameters.
- Editing Initialization Parameter Settings for an Oracle RAC Database
  You can use Enterprise Manager to view and edit the initialization parameter settings for your Oracle RAC database.
- Example: Modifying the OPEN_CURSORS Parameter
  This topic provides an example of modifying the OPEN_CURSORS parameter.

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about initialization parameters in an Oracle RAC environment

6.7.1 About Configuring Initialization Parameters for an Oracle RAC Database

Oracle Database stores initialization parameters in either a server parameter file or a text-based parameter file.

A server parameter file (SPFILE) is a type of repository for initialization parameters that is maintained on the server where the Oracle database server runs, or on shared storage for an Oracle RAC database. Initialization parameters stored in a server parameter file are persistent, in that any changes made to the parameters while an instance is running can persist across instance shutdown and startup.

An initialization parameter file (PFILE) is a text file that contains initialization parameter settings. In contrast to the SPFILE, this parameter file is not binary and does not need to be located on the database server. The text-based initialization parameter file can be read by the database, but it is not written to by the database.

By default, Oracle Database sets most parameters to a default value and this value is the same across all instances. However, many initialization parameters can also have different values on different instances as described in *Oracle Database Reference* Other parameters *must* be either unique or identical across instances, as described in the following sections:

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about initialization parameters in Oracle RAC environments

6.7.2 About Modifying the SERVICE_NAMES Parameter for Oracle RAC

The `SERVICE_NAMES` parameter should not be modified in the same manner as other initialization parameters.

The `SERVICE_NAMES` initialization parameter specifies one or more names by which clients can connect to the instance. The instance registers its service names with the listener. When a client requests a service, the listener determines which instances offer the requested service and routes the client to the appropriate instance.

In an Oracle RAC database, you should *not* modify this parameter directly. Instead, define services for your database and database instances using the Clustered Managed Database Services page in Oracle Enterprise Manager. If you must change a service, then you can use either Enterprise Manager or SRVCTL. When you use either Oracle Enterprise Manager or SRVCTL to create and start the service, the `SERVICE_NAMES` parameter is updated automatically once the service is active.

Related Topics

- About Oracle Database Services
  Services divide workloads into mutually disjoint groupings. Each service represents a workload with common attributes, service-level thresholds, and priorities.

6.7.3 About the Server Parameter File for Oracle Real Application Clusters

A server parameter file is a repository for initialization parameters.

When you create the database, Oracle creates an SPFILE in the file location that you specify. This location can be an Oracle ASM disk group or a file on a cluster file system. Oracle recommends that you store the SPFILE using Oracle ASM.

All instances in the cluster database use the same SPFILE at startup. Oracle RAC uses a traditional parameter file only if an SPFILE does not exist or if you specify `PFILE` in your `STARTUP` command. Oracle recommends that you use an SPFILE to simplify administration, maintain parameter setting consistency, and to guarantee parameter setting persistence across database shutdown and startup events. In addition, you can configure RMAN to back up your SPFILE.

> **See Also:**
>
> - *Oracle Database Backup and Recovery User’s Guide* for information about using RMAN to backup the SPFILE

6.7.4 Editing Initialization Parameter Settings for an Oracle RAC Database

You can use Enterprise Manager to view and edit the initialization parameter settings for your Oracle RAC database.

You can set the parameters on the **Current** subpage of the Initialization Parameters page to particular values to initialize many of the memory and process settings of an Oracle instance. When you modify initialization parameters using the Current tab, the changes are applied only to the running instances, not the SPFILE, unless the **Apply changes in current running instance(s) mode to SPFile** option is selected.

When you modify initialization parameters using the **SPFile** subpage, the changes are applied only to the SPFILE, not the currently running instances, unless the **Apply changes in SPFile mode to the current running instance(s)** option is selected. Resetting parameters using the SPFile subpage is different than resetting the same parameters using the Current subpage. You can either reset the parameter value for an instance back to the default value for all instances, or you can delete the default parameter setting (unset the parameter) for all instances.

To view or modify the initialization parameters using Enterprise Manager:

1. Access the Cluster Database Home page.

See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Administration** menu, select **Initialization Parameters**.

   If the Database Login page appears, then log in as a SYSDBA user, for example, SYS or SYSTEM.

3. In the Initialization Parameters page, select either the **Current** or **SPFile** subpage to modify the parameter settings.
4. Choose a parameter using the **Select** column.
5. Click **Add**, then edit the entry to change the target instance or parameter value. You can optionally add a comment describing the reason for the change.
6. After you make changes to one or more of the parameters, click **Apply** to accept and implement the changes.

Related Topics

- *Oracle Database 2 Day DBA*

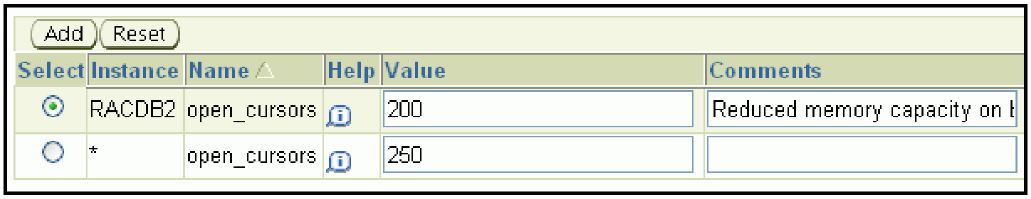
6.7.5 Example: Modifying the OPEN_CURSORS Parameter

This topic provides an example of modifying the OPEN_CURSORS parameter.

Suppose that the `open_cursors` parameter has two entries in the SPFILE:

```
*.open_cursors = 250
RACDB2.open_cursors = 200
```

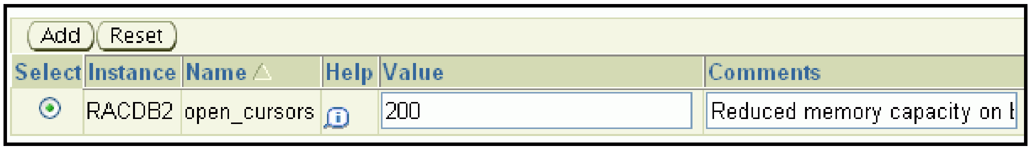
Figure 6-1 Initial Values for the OPEN_CURSORS Parameter



Select	Instance	Name △	Help	Value	Comments
●	RACDB2	open_cursors	ⓘ	200	Reduced memory capacity on t
○	*	open_cursors	ⓘ	250	

Using the Initialization Parameters page with the SPFile tab selected, if you click **Reset** for `*.open_cursors`, then Enterprise Manager deletes that entry from both the SPFILE and the displayed list of parameters, leaving only `RACDB2.open_cursors = 200` displayed.

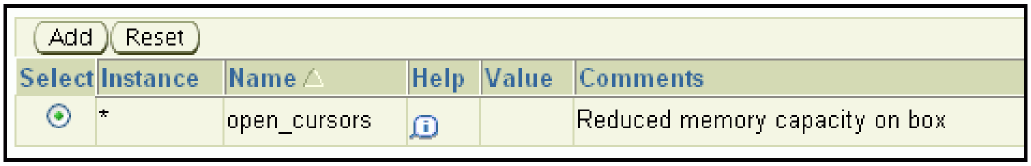
Figure 6-2 Parameter Values After Deleting Default Entries



Select	Instance	Name △	Help	Value	Comments
●	RACDB2	open_cursors	ⓘ	200	Reduced memory capacity on t

If you click **Reset** for `RACDB2.open_cursors`, then Enterprise Manager also deletes this parameter entry from both the SPFILE and the displayed list of parameters, but then a new entry, `*.open_cursors = <NULL>` is added to the displayed list of parameters for the reset parameter.

Figure 6-3   Parameter Values After Deleting Custom Entry and Default Entry



Select	Instance	Name △	Help	Value	Comments
◉	*	open_cursors	ⓘ		Reduced memory capacity on box

6.8 Limiting CPU Usage of Oracle RAC Database Instances

Oracle Database provides a method for managing CPU allocations on a multi-CPU server running multiple database instances. This method is called instance caging.

You might decide to run multiple Oracle database instances on a single multi-CPU server. When running multiple instances on a single server, the instances compete for CPU. One resource-intensive database instance could significantly degrade the performance of the other instances. For example, on a 16-CPU system with four database instances, the operating system might be running one database instance on the majority of the CPUs during a period of heavy load for that instance. This could degrade performance in the other three instances. CPU allocation decisions such as this are made solely by the operating system; the user generally has no control over them. With Instance Caging, users can partition CPU resources among multiple instances running on a server to ensure predictable performance.

A simple way to limit CPU consumption for each database instance is to use instance caging. Instance caging is a method that uses an initialization parameter to limit the number of CPUs that an instance can use simultaneously. In the previous example, if you use instance caging to limit the number of CPUs to four for each of the four instances, there is less likelihood that one instance can interfere with the others. When constrained to four CPUs, an instance might become CPU-bound. This is when the Resource Manager begins to do its work to allocate CPU among the various database sessions according to the resource plan that you set for the instance. Thus, instance caging and the Resource Manager together provide a simple, effective way to manage multiple instances on a single server.

To enable instance caging, do the following for each instance on the server:

1. Enable the Resource Manager by assigning a resource plan, and ensure that the resource plan has CPU directives, using the `MGMT_P1` through `MGMT_P8` parameters.

   See *Oracle Database Administrator’s Guide* for instructions.

2. Set the `CPU_COUNT` initialization parameter for either all instances, or the instances on the servers that you need to constrain CPU usage.

   If you enable instance caging and set a utilization limit in your resource plan, then the absolute limit is computed as a percentage of the allocated CPU resources. For example, if you enable instance caging and set the `CPU_COUNT` to 4, and a consumer group has a utilization limit of 50%, then the consumer group can use a maximum of 50% of 4 CPUs, which is 2 CPUs.

6.9 About Administering Storage in Oracle RAC

This topic provides additional information for using Enterprise Manager to manage some storage structures of an Oracle RAC database.

- About Automatic Undo Management in Oracle RAC
  Oracle RAC automatically manages undo segments within a specific undo tablespace that is assigned to an instance.
- Oracle Automatic Storage Management in Oracle RAC
  Oracle Automatic Storage Management (Oracle ASM) automatically optimizes storage to maximize performance by managing the storage configuration across the disks that Oracle ASM manages.
- Administering Redo Logs in Oracle RAC
  These topics provide an overview of additional concepts and procedures for configuring redo log files in Oracle RAC environments.

> **See Also:**
>
> *Oracle Database 2 Day DBA* for managing storage for single-instance databases

6.9.1 About Automatic Undo Management in Oracle RAC

Oracle RAC automatically manages undo segments within a specific undo tablespace that is assigned to an instance.

Only the instance assigned to the undo tablespace can modify the contents of that tablespace. However, each instance can read the undo data blocks created by any instance. Also, when performing transaction recovery, any instance can update any undo tablespace, if that undo tablespace is not currently being used by another instance for undo generation or transaction recovery. You assign undo tablespaces in your Oracle RAC database by specifying a different value for the `UNDO_TABLESPACE` parameter for each instance in your SPFILE or individual PFILEs. You cannot simultaneously use automatic undo management and manual undo management in an Oracle RAC database. In other words, all instances of an Oracle RAC database must operate in the same undo mode.

> **See Also:**
>
> *Oracle Database 2 Day DBA* for more information about managing the undo data for your database

6.9.2 Oracle Automatic Storage Management in Oracle RAC

Oracle Automatic Storage Management (Oracle ASM) automatically optimizes storage to maximize performance by managing the storage configuration across the disks that Oracle ASM manages.

Oracle ASM optimizes storage by evenly distributing the storage load across all the available storage within your cluster database environment. Oracle ASM partitions your total disk space requirements into uniformly sized units across all the disks in a disk group. Oracle ASM can also automatically mirror data to prevent data loss. Because of these features, Oracle ASM also significantly reduces your administrative overhead.

As in single-instance Oracle databases, using Oracle ASM in Oracle RAC does not require I/O tuning. The following topics describe Oracle ASM and Oracle ASM administration:

- About Oracle ASM Components in Oracle RAC
  An Oracle ASM instance can operate in several configurations in Oracle Flex ASM.
- About Disk Group Configurations for Oracle ASM in Oracle RAC
  When you create a disk group for a cluster, or add new disks to an existing clustered disk group, you must prepare only the underlying physical storage on shared disks.

6.9.2.1 About Oracle ASM Components in Oracle RAC

An Oracle ASM instance can operate in several configurations in Oracle Flex ASM.

Oracle Flex ASM enables an Oracle ASM instance to run on a separate physical server from the database servers. With this deployment, larger clusters of Oracle ASM instances can support more database clients while reducing the Oracle ASM footprint for the overall system. When using Oracle Flex ASM, Oracle ASM clients are configured with direct access to storage.

With Oracle Flex ASM, you can consolidate all the storage requirements into a single set of disk groups. All these disk groups are mounted by and managed by a small set of Oracle ASM instances running in a single cluster. You can specify the number of Oracle ASM instances with a cardinality setting. The default is three instances.

Each Oracle ASM instance has either an SPFILE or PFILE type parameter file.

For the environment described in this guide, there is an Oracle ASM instance on each node and the Oracle ASM instances use SPFILEs.

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*

6.9.2.2 About Disk Group Configurations for Oracle ASM in Oracle RAC

When you create a disk group for a cluster, or add new disks to an existing clustered disk group, you must prepare only the underlying physical storage on shared disks.

The shared disk requirement is the only substantial difference between using Oracle ASM in an Oracle RAC database compared to using it in a single-instance Oracle database. Oracle ASM automatically rebalances the storage load after you add or delete a disk or disk group.

In a cluster, each Oracle ASM instance manages the metadata updates to the disk groups for the node on which it is running. In addition, each Oracle ASM instance coordinates disk group metadata with other nodes in the cluster.

As in single-instance Oracle databases, you can use Enterprise Manager, Oracle ASM configuration assistant (ASMCA), Oracle ASM command-line utility (ASMCMD), SQL*Plus, and SRVCTL to administer disk groups for Oracle ASM in an Oracle RAC environment.

> **See Also:**
>
> - About Oracle ASM Running in an Oracle RAC Environment for an illustration of the relationship between Oracle RAC instances, Oracle ASM instances, and the shared storage
> - *Oracle Automatic Storage Management Administrator's Guide* for information about how to use SQL*Plus to administer Oracle ASM instances

6.9.3 Administering Redo Logs in Oracle RAC

These topics provide an overview of additional concepts and procedures for configuring redo log files in Oracle RAC environments.

Managing redo log files in Oracle RAC environments is similar to managing redo log files in single-instance Oracle Database environments.

- About Redo Log Groups and Redo Threads in Oracle RAC Databases
  Redo logs contain a record of changes that have been made to data files.
- About Accessing Redo Log Files for an Oracle RAC Database
  In an Oracle RAC database, each instance writes and archives the redo log groups in its redo thread in the same manner that single-instance Oracle databases do.
- Using Enterprise Manager to View and Create Online Redo Log Files
  On the Redo Log Groups page, you can create additional redo log groups and add members to the redo log group. The Thread column identifies the instance, or redo thread, to which a redo log file belongs.

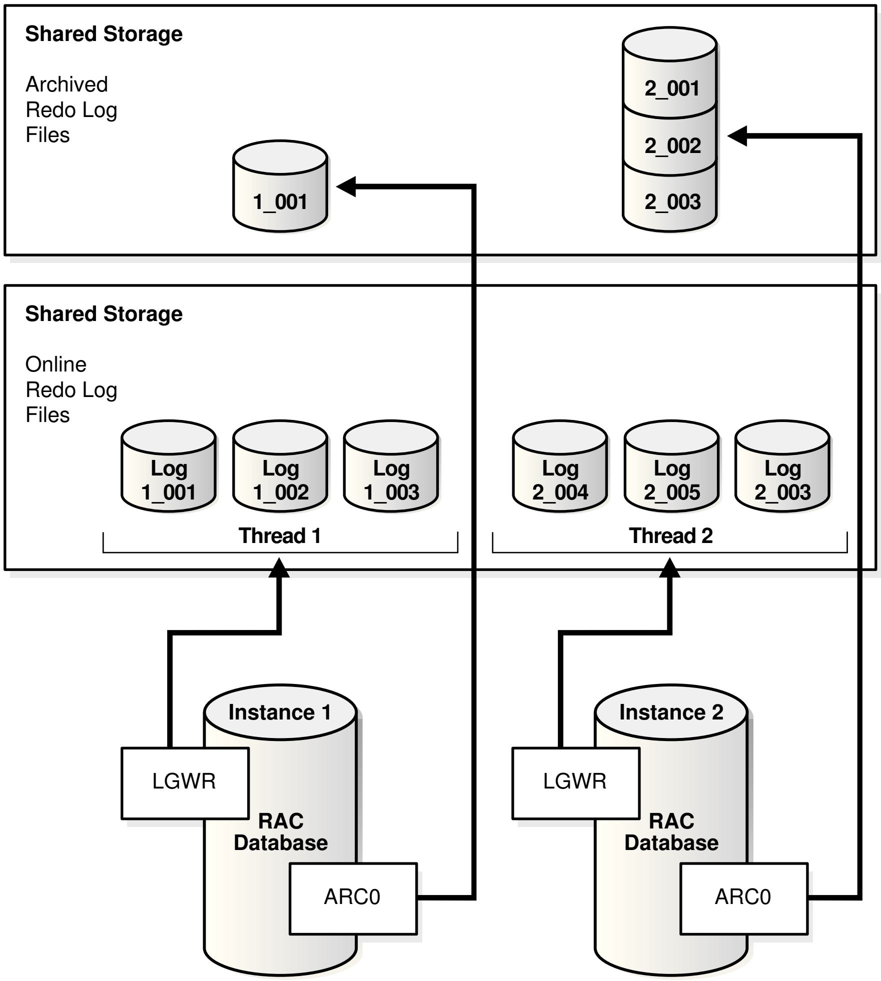
> **See Also:**
>
> - *Oracle Database 2 Day DBA* for more information about online redo logs
> - *Oracle Automatic Storage Management Administrator's Guide* for information about creating online redo logs groups in Oracle ASM

6.9.3.1 About Redo Log Groups and Redo Threads in Oracle RAC Databases

Redo logs contain a record of changes that have been made to data files.

In a single-instance Oracle database, redo logs are stored in two or more redo log file groups. Each of these groups contains a redo log file and possibly one or more mirrored copies of that file. In an Oracle RAC database, each instance requires its own set of redo log groups, which is known as a redo thread. Mirrored copies of the redo log files provide your system with extra protection against data loss that is due to hardware failures or data corruption. If a redo log file is unreadable, then Oracle Database attempts to access its mirrored copy. The redo log file mirrors should be located on different disk devices from the primary redo log files.

Figure 6-4 Illustration of Redo Threads for an Oracle RAC Database



Each instance's redo thread must contain at least two redo log groups. Each redo log group should contain at least two members: a redo log and its mirrored copy. If you create your Oracle RAC database using DBCA, then your Oracle RAC database automatically implements a configuration that meets the Oracle recommendations.

You should create redo log groups only if you are using administrator-managed databases. For policy-managed databases, if an instance starts due to a change in server pool cardinality, then Oracle Database automatically creates redo log files, enables a redo thread for the instance if there is not a redo thread allocated to that instance, and creates the undo tablespace if there is not an undo tablespace allocated to that instance. The database must be using Oracle Managed Files and Oracle ASM in this situation.

In an Oracle RAC database, all the redo log files reside on shared storage. In addition, each instance must have access to the redo log files of all the other instances in the cluster. If your Oracle RAC database uses Oracle ASM, then Oracle ASM manages the shared storage for the redo log files and the access to those files.

> **Note:**
>
> DBCA creates redo threads and undo tablespaces up to the maximum value specified by the `CLUSTER_DATABASE_INSTANCES` parameter at the time of creation for better performance.

> **See Also:**
>
> - *Oracle Real Application Clusters Administration and Deployment Guide* for more information about redo log file storage in Oracle RAC.

6.9.3.2 About Accessing Redo Log Files for an Oracle RAC Database

In an Oracle RAC database, each instance writes and archives the redo log groups in its redo thread in the same manner that single-instance Oracle databases do.

However, in recovery mode, the instance performing the recovery can read and process all the redo threads for the database, regardless of which instance generated the redo thread. Being able to read all the redo threads enables a running instance to recover the work completed by one or more failed instances.

In case of instance failure, a surviving instance can read the redo logs of the failed instance. Users can continue to access and update the database without waiting for the failed instance to be restarted. For example, assume that you have an Oracle RAC database with two instances, instance A and instance B. If instance A is down, then instance B can read the redo log files for both instance A and B to ensure a successful recovery.

6.9.3.3 Using Enterprise Manager to View and Create Online Redo Log Files

On the Redo Log Groups page, you can create additional redo log groups and add members to the redo log group. The Thread column identifies the instance, or redo thread, to which a redo log file belongs.

To access the redo log file groups using Enterprise Manager:

1. Access the Cluster Database Home page.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Administration** menu, select **Storage**, and then select **Redo Log Groups**.

   The Redo Log Groups page appears.

3. On the Redo Log Group page, perform any of the following tasks:

   - View the status, number of members, thread, file size, status, and archive status of each redo log group
   - Create or delete log groups
   - Edit a redo log group to add or remove members
   - Perform other redo log group management tasks, including clearing log files, duplicating a log group, generating sizing advice, and forcing a log switch

> **See Also:**
>
> - *Oracle Real Application Clusters Administration and Deployment Guide* for additional information about redo threads in an Oracle RAC environment
> - *Oracle Automatic Storage Management Administrator's Guide* for information about creating online redo logs groups in Oracle ASM
> - *Oracle Database 2 Day DBA* for more information about managing online redo log files

7 Administering Oracle Clusterware Components

The main components to manage in your Oracle Clusterware environment are the voting disks and the Oracle Cluster Registry (OCR).

- About Oracle Clusterware
  Oracle Clusterware includes a high availability framework for managing any application that runs on your cluster. Oracle Real Application Clusters (Oracle RAC) uses Oracle Clusterware as the infrastructure that binds multiple nodes that then operate as a single server.
- Managing the Oracle Clusterware Stack
  By default, Oracle Clusterware is configured to restart whenever the server it resides on is restarted. During certain maintenance operations, you may be required to stop or start the Oracle Clusterware stack manually.
- Administering Voting Disks for Oracle Clusterware
  Voting disks manage information about node membership. Each voting disk must be accessible by all nodes in the cluster for nodes to be members of the cluster.
- Backing Up and Recovering the Oracle Cluster Registry
  Oracle Clusterware automatically creates Oracle Cluster Registry (OCR) backups every four hours. At any one time, Oracle Clusterware always retains the latest three backup copies of the OCR that are four hours old, one day old, and one week old.
- Changing the Oracle Cluster Registry Configuration
  To ensure proper operation of Oracle Clusterware, make sure the OCR is highly available.
- Troubleshooting the Oracle Cluster Registry
  The Oracle Cluster Registry (OCR) manages Oracle Clusterware and Oracle RAC database configuration information.

> **See Also:**
>
> - "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"

7.1 About Oracle Clusterware

Oracle Clusterware includes a high availability framework for managing any application that runs on your cluster. Oracle Real Application Clusters (Oracle RAC) uses Oracle Clusterware as the infrastructure that binds multiple nodes that then operate as a single server.

In an Oracle RAC environment, Oracle Clusterware monitors all Oracle components (such as instances and listeners). If a failure occurs, then Oracle Clusterware automatically attempts to restart the failed component and also redirects operations to a surviving component.

Oracle Clusterware manages applications to ensure they start when the system starts. Oracle Clusterware also monitors the applications to ensure they are always available. For example, if an application process fails, then Oracle Clusterware attempts to restart the process based on scripts that you customize. If a node in the cluster fails, then you can program application processes that typically run on the failed node to restart on another node in the cluster.

- About the Voting Disks
  Voting disks are one of the key files for Oracle Clusterware.
- About Oracle Cluster Registry
  The Oracle Cluster Registry is a file that contains information about the cluster node list and instance-to-node mapping information.
- About High Availability of Oracle Clusterware Files
  High availability configurations have redundant hardware and software that maintain operations by avoiding single points of failure.

Related Topics

- About Oracle Grid Infrastructure for a Cluster and Oracle RAC
  Oracle Grid Infrastructure for a cluster refers to the installation of the combined products. Oracle Clusterware and Oracle ASM are still individual products, and are referred to by those names.

7.1.1 About the Voting Disks

Voting disks are one of the key files for Oracle Clusterware.

The voting disk records node membership information. A node must be able to access more than half the voting disks at any time. To avoid simultaneous loss of multiple voting disks, each voting disk should be on a storage device that does not share any components (controller, interconnect, and so on) with the storage devices used for the other voting disks.

For example, if you have five voting disks configured, then a node must be able to access at least three of the voting disks at any time. If a node cannot access the minimum required number of voting disks, then it is *evicted*, or removed, from the cluster. After the cause of the failure has been corrected and access to the voting disks has been restored, you can instruct Oracle Clusterware to recover the failed node and restore it to the cluster.

7.1.2 About Oracle Cluster Registry

The Oracle Cluster Registry is a file that contains information about the cluster node list and instance-to-node mapping information.

The Oracle Cluster Registry (OCR) also contains information about Oracle Clusterware resource profiles for resources that you have customized. The voting disk data is also backed up in OCR.

Each node in a cluster also has a local copy of the OCR, called an Oracle Local Registry (OLR), that is created when Oracle Clusterware is installed. Multiple processes on each node have simultaneous read and write access to the OLR particular to the node on which they reside, whether Oracle Clusterware is fully functional. By default, OLR is located at `*Grid_home*/cdata/*$HOSTNAME*.olr`

7.1.3 About High Availability of Oracle Clusterware Files

High availability configurations have redundant hardware and software that maintain operations by avoiding single points of failure.

When a component is down, Oracle Clusterware redirects its managed resources to a redundant component. However, if a disaster strikes, or a massive hardware failure occurs, then having redundant components might not be enough. To fully protect your system it is important to have backups of your critical files.

The Oracle Clusterware installation process creates the voting disk and the OCR on shared storage. If you select the option for normal redundant copies during the installation process, then Oracle Clusterware automatically maintains redundant copies of these files to prevent the files from becoming single points of failure. The normal redundancy feature also eliminates the need for third-party storage redundancy solutions. When you use normal redundancy, Oracle Clusterware automatically maintains two copies of the OCR file and three copies of the voting disk file.

> **See Also:**
>
> *Oracle Clusterware Administration and Deployment Guide* for more information about managing voting disks

7.2 Managing the Oracle Clusterware Stack

By default, Oracle Clusterware is configured to restart whenever the server it resides on is restarted. During certain maintenance operations, you may be required to stop or start the Oracle Clusterware stack manually.

> **Note:**
>
> Do not use Oracle Clusterware Control (CRSCTL) commands on Oracle entities (such as resources, resource types, and server pools) that have names beginning with `ora` unless you are directed to do so by Oracle Support. The Server Control utility (SRVCTL) is the correct utility to use on Oracle entities.

- Starting Oracle Clusterware
  You use the CRSCTL utility to manage Oracle Clusterware.
- Stopping Oracle Clusterware
  You use the CRSCTL utility to manage Oracle Clusterware.

See Also:

- "About Oracle Clusterware"

7.2.1 Starting Oracle Clusterware

You use the CRSCTL utility to manage Oracle Clusterware.

- If the Oracle High Availability Services daemon (OHASD) is running on all the cluster nodes, then you can start the entire Oracle Clusterware stack (all the processes and resources managed by Oracle Clusterware) on all nodes in the cluster by executing the following command on any node:

  ```
  crsctl start cluster -all
  ```

- You can start the Oracle Clusterware stack on specific nodes by using the `-n` option followed by a space-delimited list of node names, for example:

  ```
  crsctl start cluster -n racnode1 racnode4
  ```

  To use the `crsctl start cluster` command, the OHASD process must be running on the specified nodes.

- To start the entire Oracle Clusterware stack on a node, including the OHASD process, run the following command on that node:

  ```
  crsctl start crs
  ```

See Also:

- "About Oracle Clusterware"
- "Managing the Oracle Clusterware Stack"

7.2.2 Stopping Oracle Clusterware

You use the CRSCTL utility to manage Oracle Clusterware.

- To stop Oracle Clusterware on all nodes in the cluster, execute the following command on any node:

  ```
  crsctl stop cluster -all
  ```

  The previous command stops the resources managed by Oracle Clusterware, the Oracle ASM instance, and all the Oracle Clusterware processes (except for OHASD and its dependent processes).

- To stop Oracle Clusterware and Oracle ASM on select nodes, include the `-n` option followed by a space-delimited list of node names, for example:

  ```
  crsctl stop cluster -n racnode1 racnode3
  ```

If you do not include either the `-all` or the `-n` option in the `stop cluster` command, then Oracle Clusterware and its managed resources are stopped only on the node where you execute the command.

- To completely shut down the entire Oracle Clusterware stack, including the OHASD process, use the `crsctl stop crs` command.

  CRSCTL attempts to gracefully stop the resources managed by Oracle Clusterware during the shutdown of the Oracle Clusterware stack. If any resources that Oracle Clusterware manages are still running after executing the `crsctl stop crs` command, then the command fails. You must then use the `-f` option to unconditionally stop all resources and stop the Oracle Clusterware stack, for example:

  ```
  crsctl stop crs -f
  ```

  > **Note:**
  >
  > When you shut down the Oracle Clusterware stack, you also shut down the Oracle Automatic Storage Management (Oracle ASM) instances. If the Oracle Clusterware files (voting disk and OCR) are stored in an Oracle ASM disk group, then the only way to shut down the Oracle ASM instances is to shut down the Oracle Clusterware stack.

> **See Also:**
>
> - "About Oracle Clusterware"
> - "Managing the Oracle Clusterware Stack"

7.3 Administering Voting Disks for Oracle Clusterware

Voting disks manage information about node membership. Each voting disk must be accessible by all nodes in the cluster for nodes to be members of the cluster.

- Adding and Removing Voting Disks
  You must manually add and remove voting disks if they are stored on local storage.
- Backing Up and Recovering Voting Disks
  You do not have to back up the voting disk. The voting disk data is automatically backed up in the Oracle Cluster Registry (OCR) as part of any configuration change and is automatically restored to any voting disk added.
- Migrating Voting Disks to Oracle ASM Storage
  You should store the Oracle Clusterware voting disk files in an Oracle ASM disk group.

7.3.1 Adding and Removing Voting Disks

You must manually add and remove voting disks if they are stored on local storage.

If you choose to store Oracle Clusterware files on Oracle ASM and use redundancy for the disk group, then Oracle ASM automatically maintains the ideal number of voting files based on the redundancy of the disk group.

If you use a different form of shared storage to store the voting disks, then you can dynamically add and remove voting disks after installing Oracle RAC. Do this using the following commands where *path* is the fully qualified path for the additional voting disk.

To add or remove a voting disk that is stored on disk:

1. Run the following command as the `grid` user to add a voting disk:

   ```
   crsctl add css votedisk path
   ```

2. Run the following command as the `grid` user to remove a voting disk:

   ```
   crsctl delete css votedisk path
   ```

7.3.2 Backing Up and Recovering Voting Disks

You do not have to back up the voting disk. The voting disk data is automatically backed up in the Oracle Cluster Registry (OCR) as part of any configuration change and is automatically restored to any voting disk added.

- Backing Up Voting Disks
  The voting disk data is automatically backed up in Oracle Cluster Registry (OCR) as part of any configuration change so you do not have to perform manual backups of the voting disk.
- Replacing Voting Disks
  You replace a voting disk by deleting the unusable voting disk and then adding a new voting disk to your configuration.
- Restoring Voting Disks
  If all voting disks are corrupted, then you can restore them

> **See Also:**
>
> - "About Oracle Clusterware"
> - *Oracle Clusterware Administration and Deployment Guide* for more information about managing voting disks

7.3.2.1 Backing Up Voting Disks

The voting disk data is automatically backed up in Oracle Cluster Registry (OCR) as part of any configuration change so you do not have to perform manual backups of the voting disk.

The voting disk data is automatically backed up in OCR as part of any configuration change so you do not have to perform manual backups of the voting disk. The voting disk files are backed up automatically by Oracle Clusterware if the contents of the files have changed in the following ways:

- Configuration parameters, for example `misscount`, have been added or modified
- After performing voting disk `add` or `delete` operations

See Also:

Oracle Clusterware Administration and Deployment Guide for more information about managing voting disks

7.3.2.2 Replacing Voting Disks

You replace a voting disk by deleting the unusable voting disk and then adding a new voting disk to your configuration.

If a voting disk is damaged, and no longer usable by Oracle Clusterware, then you can replace or re-create the voting disk. The voting disk contents are restored from a backup when a new voting file is added; this occurs regardless of whether the voting disk file is stored in Oracle Automatic Storage Management (Oracle ASM).

To replace a corrupt, damaged, or missing voting disk that is not stored in Oracle ASM:

1. Use CRSCTL to remove the damaged voting disk. For example, if the damaged voting disk is stored in the disk location `/dev/sda3`, then execute the command:

   ```
   crsctl delete css votedisk /dev/sda3
   ```

2. Use CRSCTL to create a new voting disk in the same location, for example:

   ```
   crsctl add css votedisk /dev/sda3
   ```

See Also:

Oracle Clusterware Administration and Deployment Guide for more information about managing voting disks

7.3.2.3 Restoring Voting Disks

If all voting disks are corrupted, then you can restore them

- The process of restoring voting disks is described in *Oracle Clusterware Administration and Deployment Guide*.

> **Note:**
>
> Restoring a voting disk from a copy created with the Linux or UNIX operating system `dd` command is not supported

7.3.3 Migrating Voting Disks to Oracle ASM Storage

You should store the Oracle Clusterware voting disk files in an Oracle ASM disk group.

If you choose to store your voting disks in Oracle ASM, then Oracle ASM stores all the voting disks for the cluster in the disk group you choose. You cannot combine voting disks stored in Oracle ASM and voting disks not stored in Oracle ASM in the same cluster.

The number of voting files you can store in a particular Oracle ASM disk group depends upon the redundancy of the disk group. By default, Oracle ASM puts each voting disk in its own failure group within the disk group. A normal redundancy disk group must contain at least two failure groups but if you are storing your voting disks on Oracle ASM, then a normal redundancy disk group must contain at least three failure groups. A high redundancy disk group must contain at least three failure groups.

Once you configure voting disks on Oracle ASM, you can only make changes to the voting disks' configuration using the `crsctl replace votedisk` command. This is true even in cases where there are no working voting disks. Despite the fact that the `crsctl query css votedisk` command reports zero voting disks in use, Oracle Clusterware remembers the fact that Oracle ASM was in use and the `replace` verb is required. Only after you use the `replace` verb to move voting disks back to non-Oracle ASM storage are the CRSCTL commands `add css votedisk` and `delete css votedisk` again usable.

To move voting disks from shared storage to an Oracle ASM disk group:

1. Use the Oracle ASM Configuration Assistant (ASMCA) to create an Oracle ASM disk group.
2. Verify that the ASM Compatibility attribute for the disk group is set to 12.1.0.0 or higher.
3. Use CRSCTL to create a voting disk in the Oracle ASM disk group by specifying the disk group name in the following command:

   ```
   crsctl replace votedisk +ASM_disk_group
   ```

> **See Also:**
>
> - "Creating Additional Oracle ASM Disk Groups"
> - *Oracle Automatic Storage Management Administrator's Guide* for more information about disk group compatibility attributes
> - *Oracle Clusterware Administration and Deployment Guide* for more information about managing voting disks

7.4 Backing Up and Recovering the Oracle Cluster Registry

Oracle Clusterware automatically creates Oracle Cluster Registry (OCR) backups every four hours. At any one time, Oracle Clusterware always retains the latest three backup copies of the OCR that are four hours old, one day old, and one week old.

You cannot customize the backup frequencies or the number of files that Oracle Clusterware retains. You can use any backup software to copy the automatically generated backup files at least once daily to a different device from where the primary OCR file resides.

- Viewing Available OCR Backups
  Use the `ocrconfig` utility to view the backups generated automatically by Oracle Clusterware.
- Manually Backing Up the OCR
  Use the `ocrconfig` utility to force Oracle Clusterware to perform a backup of Oracle Cluster Registry (OCR) at any time, rather than wait for the automatic backup that occurs at four-hour intervals.
- Recovering the OCR
  There are two methods for recovering the OCR. The first method uses automatically generated OCR file copies and the second method uses manually created OCR export files.

7.4.1 Viewing Available OCR Backups

Use the `ocrconfig` utility to view the backups generated automatically by Oracle Clusterware.

To find the most recent backup of the OCR:

- Run the following command on any node in the cluster:

  ```
  ocrconfig -showbackup
  ```

7.4.2 Manually Backing Up the OCR

Use the `ocrconfig` utility to force Oracle Clusterware to perform a backup of Oracle Cluster Registry (OCR) at any time, rather than wait for the automatic backup that occurs at four-hour intervals.

This option is especially useful when you want to obtain a binary backup on demand, such as before you make changes to OCR. The Oracle Local Registry (OLR) only supports manual backups.

To manually backup the contents of the OCR:

1. Log in as the `root` user.
2. Use the following command to force Oracle Clusterware to perform an immediate backup of the OCR:

   ```
   ocrconfig -manualbackup
   ```

   The date and identifier of the recently generated OCR backup is displayed.

3. If you must change the location for the OCR backup files, then use the following command, where `directory_name` is the new location for the backups:

   ```
   ocrconfig -backuploc directory_name
   ```

The default location for generating backups on Oracle Linux systems is `Grid_home/cdata/cluster_name` where *cluster_name* is the name of your cluster and *Grid_home* is the home directory of the Oracle Grid Infrastructure for a cluster installation. Because the default backup is on a local file system, Oracle recommends that you include the backup file created with the `ocrconfig` command as part of your operating system backup using standard operating system or third-party tools.

> **Note:**
>
> You can use the `ocrconfig -backuploc` command to change the location where the OCR backups are created.

7.4.3 Recovering the OCR

There are two methods for recovering the OCR. The first method uses automatically generated OCR file copies and the second method uses manually created OCR export files.

- Checking the Status of the OCR
  In event of a failure, before you attempt to restore the Oracle Cluster Registry (OCR), ensure that the OCR is unavailable.
- Restoring the OCR from Automatically Generated OCR Backups
  When restoring the OCR from automatically generated backups, you first have to determine which backup file to use for the recovery.

7.4.3.1 Checking the Status of the OCR

In event of a failure, before you attempt to restore the Oracle Cluster Registry (OCR), ensure that the OCR is unavailable.

To check the status of the OCR:

1. Run the following command:

   ```
   ocrcheck
   ```

2. If this command does not display the message `'Device/File integrity check succeeded'` for at least one copy of the OCR, then all copies of the OCR have failed. You must restore the OCR from a backup or OCR export.
3. If there is at least one copy of the OCR available, then you can use that copy to restore the other copies of the OCR.

> **See Also:**
>
> - "About the OCRCHECK Utility"

7.4.3.2 Restoring the OCR from Automatically Generated OCR Backups

When restoring the OCR from automatically generated backups, you first have to determine which backup file to use for the recovery.

To restore the OCR from an automatically generated backup on an Oracle Linux system:

1. Log in as the `root` user.
2. Identify the available OCR backups using the `ocrconfig` command:

   ```
   [root]# ocrconfig -showbackup
   ```

3. Review the contents of the backup using the following `ocrdump` command, where *file_name* is the name of the OCR backup file for which the contents should be written out to the file *ocr_dump_output_file*:

   ```
   [root]# ocrdump ocr_dump_output_file -backupfile file_name
   ```

   If you do not specify an output file name, then the utility writes the OCR contents to a file named `OCRDUMPFILE` in the current directory.

4. As the `root` user, stop Oracle Clusterware on all the nodes in your cluster by executing the following command:

   ```
   [root]# crsctl stop cluster -all
   ```

5. As the `root` user, restore the OCR by applying an OCR backup file that you identified in Step 2 using the following command, where *file_name* is the name of the OCR to restore. Ensure that the OCR devices that you specify in the OCR configuration exist, and that these OCR devices are valid before running this command.

   ```
   [root]# ocrconfig -restore file_name
   ```

6. As the `root` user, restart Oracle Clusterware on all the nodes in your cluster by running the following command:

   ```
   [root]# crsctl start cluster -all
   ```

7. Use the Cluster Verification Utility (CVU) to verify the OCR integrity. Exit the `root` user account, and, as the software owner of the Oracle Grid Infrastructure for a cluster installation, run the following command, where the `-n all` argument retrieves a list of all the cluster nodes that are configured as part of your cluster:

   ```
   cluvfy comp ocr -n all [-verbose]
   ```

7.5 Changing the Oracle Cluster Registry Configuration

To ensure proper operation of Oracle Clusterware, make sure the OCR is highly available.

The OCR contains information about the cluster node list, which instances run on which nodes, and information about Oracle Clusterware resource profiles for applications that have been modified to be managed by Oracle Clusterware.

Note:

The operations in this section affect the OCR for the entire cluster. However, the `ocrconfig` command cannot modify OCR configuration information for nodes that are shut down or for nodes on which Oracle Clusterware is not running. Avoid shutting down nodes while modifying the OCR using the `ocrconfig` command.

- Adding an OCR Location
  You can add an Oracle Cluster Registry (OCR) location after an upgrade or after completing the Oracle RAC installation.
- Migrating the OCR to Oracle ASM Storage
  You should store the Oracle Cluster Registry (OCR) in an Oracle Automatic Storage Management (Oracle ASM) disk group.
- Replacing an OCR
  If you must change the location of an existing Oracle Cluster Registry (OCR), or change the location of a failed OCR to the location of a working one, then you can use this procedure if one OCR file remains online.
- Removing an OCR
  To remove an Oracle Cluster Registry (OCR) file, at least one copy of the OCR must be online.
- Repairing an OCR Configuration on a Local Node
  If a node in your cluster was not available when you modified the OCR configuration, then you must repair the OCR configuration on that node before it is restarted.

7.5.1 Adding an OCR Location

You can add an Oracle Cluster Registry (OCR) location after an upgrade or after completing the Oracle RAC installation.

Oracle Clusterware supports up to five OCR copies. Additional OCR copies provide greater fault tolerance.

To add an OCR file:

- As the `root` user, enter the following command to add a new OCR file:

  ```
  [root]# ocrconfig -add new_ocr_file_name
  ```

  This command updates the OCR configuration on all the nodes on which Oracle Clusterware is running.

7.5.2 Migrating the OCR to Oracle ASM Storage

You should store the Oracle Cluster Registry (OCR) in an Oracle Automatic Storage Management (Oracle ASM) disk group.

To improve Oracle Clusterware storage manageability, OCR is configured, by default, to use Oracle ASM in Oracle Database 12*c*. However, if you upgrade from a previous release, then you can migrate OCR to reside on Oracle ASM, and take advantage of the improvements in managing Oracle Clusterware storage.

The OCR inherits the redundancy of the disk group. If you want high redundancy for the OCR, then you must create an Oracle ASM disk group with high redundancy. You should use a disk group with at least normal redundancy, unless you have an external mirroring solution. If you store the OCR in an Oracle ASM disk group, and the Oracle ASM instance fails on a node, then the OCR becomes unavailable only on that node. The failure of one Oracle ASM instance does not affect the availability of the entire cluster.

Oracle does not support storing the OCR on different storage types simultaneously, such as storing the OCR on both Oracle ASM and a shared file system, except during a migration. After you have migrated the OCR to Oracle ASM storage, you must delete the existing OCR files.

To move the OCR from shared storage to an Oracle ASM disk group:

1. Ensure the upgrade to Oracle Clusterware 12c is complete. Run the following command to verify the current running version:

   ```
   $ crsctl query crs activeversion
   ```

2. Verify that the ASM Compatibility attribute for the disk group is set to 11.2.0.2 or higher.
3. Use the Oracle ASM Configuration Assistant (ASMCA) to create an Oracle ASM disk group that is at least the same size as the existing OCR and has at least normal redundancy.

   Ensure that Oracle ASM disk groups that you create are mounted on all of the nodes in the cluster.
4. To add OCR to an Oracle ASM disk group, ensure that the Oracle Clusterware stack is running, then execute the following OCRCONFIG command as the `root` user, specifying the Oracle ASM disk group name:

   ```
   # ocrconfig -add +ASM_disk_group
   ```

   You can run this command more than once if you add multiple OCR locations. You can have up to five OCR locations. However, each successive run must point to a different disk group.
5. Remove the non-Oracle ASM storage locations by running the following command as the `root` user:

   ```
   # ocrconfig -delete old_storage_location
   ```

   You must run this command once for every shared storage location for the OCR that is not using Oracle ASM. For example:

   ```
   # ocrconfig -delete /ocrdata/ocr_1
   # ocrconfig -delete /ocrdata/ocr_2
   ```

See Also:

- "Creating Additional Oracle ASM Disk Groups"
- *Oracle Automatic Storage Management Administrator's Guide* for more information about disk group compatibility attributes
- *Oracle Clusterware Administration and Deployment Guide* for more information about migrating the OCR to Oracle ASM

7.5.3 Replacing an OCR

If you must change the location of an existing Oracle Cluster Registry (OCR), or change the location of a failed OCR to the location of a working one, then you can use this procedure if one OCR file remains online.

To change the location of an OCR or replace an OCR file:

1. Use the OCRCHECK utility to verify that a copy of the OCR other than the one you are going to replace is *online*, using the following command:

   ```
   ocrcheck
   ```

   > **Note:**
   >
   > The OCR that you are *replacing* can be either online or offline.

2. Use the following command to verify that Oracle Clusterware is running on the node on which you are going to perform the replace operation:

   ```
   crsctl check cluster -all
   ```

3. As the `root` user, enter the following command to designate a new location for the specified OCR file:

   ```
   [root]# ocrconfig -replace source_ocr_file -replacement destination_ocr_file
   ```

   This command updates the OCR configuration on all the nodes on which Oracle Clusterware is running.

4. Use the OCRCHECK utility to verify that OCR replacement file is online:

   ```
   ocrcheck
   ```

7.5.4 Removing an OCR

To remove an Oracle Cluster Registry (OCR) file, at least one copy of the OCR must be online.

You can remove an OCR location to reduce OCR-related overhead or to stop mirroring your OCR because you moved the OCR to a redundant storage system, such as a redundant array of independent disks (RAID).

To remove an OCR location from your cluster:

1. Use the OCRCHECK utility to ensure that at least one OCR other than the OCR that you are removing is online.

   ```
   ocrcheck
   ```

   > **Note:**
   >
   > Do *not* perform this OCR removal procedure unless there is at least one active OCR online.

2. As the `root` user, run the following command on any node in the cluster to remove a specific OCR file:

   ```
   [root]# ocrconfig -delete ocr_file_name
   ```

   This command updates the OCR configuration on all the nodes on which Oracle Clusterware is running.

7.5.5 Repairing an OCR Configuration on a Local Node

If a node in your cluster was not available when you modified the OCR configuration, then you must repair the OCR configuration on that node before it is restarted.

To repair an OCR configuration:

1. As the `root` user, run one or more of the following commands on the node on which Oracle Clusterware is stopped, depending on the number and type of changes that were made to the OCR configuration:

   ```
   [root]# ocrconfig –repair -add new_ocr_file_name
   ```

   ```
   [root]# ocrconfig –repair -delete ocr_file_name
   ```

   ```
   [root]# ocrconfig –repair -replace source_ocr_file -replacement dest_ocr_file
   ```

   These commands update the OCR configuration only on the node from which you run the command.

   > **Note:**
   >
   > You *cannot* perform these operations on a node on which the Oracle Clusterware daemon is running.

2. Restart Oracle Clusterware on the node you have just repaired.
3. As the `root` user, check the OCR configuration integrity of your cluster using the following command:

   ```
   [root]# ocrcheck
   ```

7.6 Troubleshooting the Oracle Cluster Registry

The Oracle Cluster Registry (OCR) manages Oracle Clusterware and Oracle RAC database configuration information.

- About the OCRCHECK Utility
  The OCRCHECK utility displays the data block format version used by the Oracle Cluster Registry (OCR), the available space and used space in the OCR, the ID used for the OCR, and the locations you have configured for the OCR.
- Common Oracle Cluster Registry Problems and Solutions
  The following table describes common Oracle Cluster Registry (OCR) problems and their corresponding solutions.

7.6.1 About the OCRCHECK Utility

The OCRCHECK utility displays the data block format version used by the Oracle Cluster Registry (OCR), the available space and used space in the OCR, the ID used for the OCR, and the locations you have configured for the OCR.

The OCRCHECK utility calculates a checksum for all the data blocks in all the OCRs that you have configured to verify the integrity of each block. It also returns an individual status for each OCR file and a result for the overall OCR integrity check. The following is a sample of the OCRCHECK output:

```
Status of Oracle Cluster Registry is as follows :
   Version                  :          3
   Total space (kbytes)     :     262144
   Used space (kbytes)      :      16256
   Available space (kbytes) :     245888
   ID                       :  570929253
   Device/File Name         : +CRS_DATA
                              Device/File integrity check succeeded
...
                              Decive/File not configured

   Cluster registry integrity check succeeded

   Logical corruption check succeeded
```

The OCRCHECK utility creates a log file in the following directory, where *Grid_home* is the location of the Oracle Grid Infrastructure for a cluster installation, and *hostname* is the name of the local node: *Grid_home*/log/*hostname*/client

The log files have names of the form ocrcheck_*nnnnn*.log, where *nnnnn* is the process ID of the operating session that issued the ocrcheck command.

Related Topics

- Viewing Available OCR Backups
  Use the ocrconfig utility to view the backups generated automatically by Oracle Clusterware.

7.6.2 Common Oracle Cluster Registry Problems and Solutions

The following table describes common Oracle Cluster Registry (OCR) problems and their corresponding solutions.

Table 7-1 Common OCR Problems and Solutions

Problem	Solution
The OCR is not mirrored.	Run the `ocrconfig` command with the `-add` option as described in the section "Adding an OCR Location".
A copy of the OCR has failed and you must replace it. Error messages are being reported in Enterprise Manager or the OCR log file.	Run the `ocrconfig` command with the `-replace` option as described in the section "Replacing an OCR".
OCRCHECK does not find a valid OCR, or all copies of the OCR are corrupted.	Run the `ocrconfig` command with the `-restore` option as described in the section "Restoring the OCR from Automatically Generated OCR Backups".
The OCR configuration was updated incorrectly.	Run the `ocrconfig` command with the `-repair` option as described in the section "Repairing an OCR Configuration on a Local Node".
You are experiencing a severe performance effect from updating multiple OCR files, or you want to remove an OCR file for other reasons.	Run the `ocrconfig` command with the `-delete` option as described in the section "Removing an OCR".
You want to change the location or storage type currently used by the OCR.	Run the `ocrconfig` command with the `-replace` option while Oracle Clusterware is running, as described in the section "Replacing an OCR". If some cluster nodes are down when you move the OCR, then you must run `ocrconfig -repair` on each node that was down before you start Oracle Clusterware on that node.

8 Client High Availability with Services, FAN, FCF, and Application Continuity

You can achieve client high availability and scalability in Oracle Real Application Clusters (Oracle RAC) using database services, Fast Application Notification (FAN), Fast Connection Failover (FCF), and Application Continuity. Older applications can use Transparent Application Failover (TAF).

8.1 About Continuous Service Availability with Database Services

You can deploy Oracle RAC environments to achieve continuous service in many different ways.

Applications using a clustered database generally want to load balance their workload across the cluster. Oracle Real Application Clusters (Oracle RAC) runs on Oracle Clusterware, which provides a highly available (HA) application framework. Oracle Clusterware provides the necessary service and integration points between Oracle RAC and custom enterprise applications using FAN. Across data centers Global Data Services (GDS) also provides these integration points for services and FAN.

Depending on the number of nodes and your environment's complexity and objectives, your choices for the optimal workload management and high availability configuration depend on a variety of considerations.

To implement continuous service for your applications using Oracle RAC database, use the following features:

See Also:

"Creating Services"

8.1.1 About Oracle Database Services

Services divide workloads into mutually disjoint groupings. Each service represents a workload with common attributes, service-level thresholds, and priorities.

Database services, referred to as services, are logical abstractions for managing workloads in Oracle Database. A single service can represent an application, multiple applications or a subset of a single application. For example, the Oracle E-Business suite defines a service for each responsibility, such as general ledger, accounts receivable, order entry, and so on. A service can be active on one or several instances of an Oracle RAC cluster and a single database instance can host multiple services.

> **Note:**
>
> A database service can only be offered on a single network.

Services provide the following benefits:

- Provide a single entity for managing applications that compete for the same resources
- Enable each workload to be managed as a unit
- Hide the complexity of the cluster from the client

You can use services to manage the workload for different types of work. For example, online users can use one service while batch processing can use a different service and reporting can use yet another service type.

Traditionally an Oracle database provided a single service and all users connected to the same service. A database always has this default database service that is the database name. This service cannot be modified. It should only be used for administrative tasks because it always enables you to connect to the database. The default database service should not be used for high availability because it cannot be disabled or relocated. Always use user-defined database services for your applications.

> **Note:**
>
> Do not use the default database service for application workloads. The default service is for administration purposes only. The default database service has the same name as the `DB_NAME`, `DB_UNIQUE_NAME`, or `PDBName`. The default database service cannot be enabled, disabled, or failed over, and it does not support load balancing.
>
> You must create at least one service as described in "Creating Services".

When a user or application connects to a database, Oracle recommends that you use a service for the connection. Oracle Database automatically creates one database service when the database is created. However, for more flexibility in the management of the applications connecting to the database and their workload, you should create one or more application services and specify which database instances offer the services.

You can define services for both policy-managed and administrator-managed databases.

- **Policy-managed database**: When you define services for a policy-managed database, you assign the service to a server pool where the database is running. You can define the service as either uniform (running on all instances in the server pool) or singleton (running on only one instance in the server pool).
- **Administrator-managed database**: When you define a service for an administrator-managed database, you define which instances normally support that service. These are known as the `PREFERRED` instances. You can also define

other instances to support a service if the preferred instance fails. These are known as AVAILABLE instances. A service that runs on an administrator-managed database must have at least one PREFERRED instance.

Services are integrated with the Database Resource Manager, which enables you to restrict the resources that a service uses within an instance. In addition, Oracle Scheduler jobs can run using a service, as opposed to using a specific instance.

- About Service Failover in Administrator-Managed Databases
  When configuring service failover in administrator-managed databases, you must configure preferred and available instances.
- About Service Failover in Policy-Managed Databases
  Service failover in policy-managed databases behaves differently if the service is UNIFORM or SINGLETON.
- About Automatic Starting of Services
  When you define a service, you can also define the management policy for that service.

See Also:

- "Creating Services"
- "Administering Services"
- *Oracle Database Administrator's Guide* for more information about managing application workloads with database services

8.1.1.1 About Service Failover in Administrator-Managed Databases

When configuring service failover in administrator-managed databases, you must configure preferred and available instances.

During normal operation a service can be running on any combination of the preferred and available instances, up to the cardinality of the service (the number of PREFERRED instances defined). Only at initial startup of the service will Oracle Clusterware attempt to start the service on the PREFERRED instances. If an instance fails, then the service fails over to one of the combined list of preferred and available instances that is not offering the service. You can also manually relocate the service to one of the instances in the combined list of preferred and available instances not offering the service.

If a service fails over to an available instance, then the service is not moved back to its preferred instance automatically. However, you can automate the relocation of a service to its preferred instance by using a FAN callout.

If you configure a preferred instance for a service, but do not specify at least one available instance for the service, then the service does not failover to another instance if the preferred instance fails.

Using Enterprise Manager, you can also specify an instance as Not Used. This setting means the service does not run on the instance, even if the preferred instance for the service fails.

See Also:

- "About FAN Callouts"
- FAST APPLICATION NOTIFICATION WITH ORACLE DATABASE 12C white paper for an example of a callout script in the section "Appendix D Sample Callout program"
- "Creating Services"
- *Oracle Database High Availability Overview* for more information about the high availability products and features available for Oracle Database

8.1.1.2 About Service Failover in Policy-Managed Databases

Service failover in policy-managed databases behaves differently if the service is UNIFORM or SINGLETON.

When you specify that a service is UNIFORM, Oracle Clusterware attempts to ensure that the service always runs on all the available instances for the specified server pool. If the instance fails, then the service is no longer available on that instance. If the cardinality of the server pool increases and an instance is added to the database, then the service is started on the new instance. You cannot manually relocate the service to a specific instance.

When you specify that a service is SINGLETON, Oracle Clusterware attempts to ensure that the service always runs on only one of the available instances for the specified server pool. If the instance fails, then the service fails over to a different instance in the server pool. You cannot specify which instance in the server pool the service should run on.

For SINGLETON services, if a service fails over to a new instance, then the service is not moved back to its original instance when that instance becomes available again.

See Also:

- "Creating Services"
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about automatic workload management using database services

8.1.1.3 About Automatic Starting of Services

When you define a service, you can also define the management policy for that service.

You can choose either an automatic or manual management policy:

- **Automatic**: The service always starts when the database starts.
- **Manual**: Requires you to start the service manually after the database starts.

Note:

When you use automatic services in an administrator-managed database, during planned database startup, services may start on the first instances that become available rather than their preferred instances.

Related Topics

- Creating Services
  You can create a service using Oracle Enterprise Manager or the SRVCTL utility.

8.1.2 About the Database Resource Manager

The Database Resource Manager is a database feature you can use to control the database resources allocated to users, applications, and services.

The Database Resource Manager ensures that users, applications, and services receive their share of the available database resources. The Database Resource Manager enables an Oracle RAC database running on one or more nodes to support multiple applications and mixed workloads with optimal efficiency.

The Database Resource Manager provides the ability to prioritize work within an Oracle database or your Oracle RAC environment. For example, high priority users, such as online workers, would get more resources to minimize response time, while lower priority users, such as batch jobs or reports, would get fewer resources, and could take longer to run. Database Resource Manager enables more granular control over resources.

Resources are allocated to users according to a resource plan specified by the database administrator. The following terms are used in specifying a resource plan:

- A **resource plan** specifies how the resources are to be distributed among various users based on resource consumer groups.
- **Resource consumer groups** enable the administrator to group user sessions by resource requirements. Resource consumer groups are different from user roles; one database user can have different sessions assigned to different resource consumer groups.
- **Resource allocation methods** are the methods or policies used by the Database Resource Manager when allocating for a particular resource. Resource consumer groups and resource plans use the resource allocation methods. The database provides the resource allocation methods that are available, but the DBA determines which method to use.
- **Resource plan directives** are a means of assigning consumer groups to particular plans and partitioning resources among consumer groups by specifying parameters for each resource allocation method.
- **Subplans**, which the DBA can create within a resource plan, enable further subdivision of resources among different users of an application.
- **Levels** provide a mechanism to specify distribution of unused resources among available users. Up to eight levels of resource allocation can be specified.

The Database Resource Manager enables you to map a resource consumer group to a service so that users who connect using that service are members of the specified resource consumer group, and thus restricted to the resources available to that resource consumer group.

You can access a tutorial for Resource Manager in Enterprise Manager. Simply navigate to the Cluster Database Home page, then, from the **Administration** menu, select **Resource Manager**, and then **Getting Started**. See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

See Also:

Oracle Database Administrator's Guide for more information about the Database Resource Manager

8.1.3 About Oracle RAC High Availability Framework

The Oracle RAC high availability framework enables Oracle RAC to maintain the database, components, and applications in a running state always.

If an instance, component, or application fails, then it can be automatically restarted to keep Oracle Database operating at full capacity.

Oracle Database focuses on maintaining service availability. In Oracle RAC, Oracle services are designed to be continuously available with workloads shared across one or more instances. The Oracle RAC high availability framework maintains service availability by storing the configuration information for each service in the Oracle Cluster Registry (OCR). Oracle Clusterware recovers and balances services across instances based on the service definition.

See Also:

Oracle Database High Availability Overview for information about determining the high availability requirements for your system

8.1.4 About Fast Application Notification (FAN)

Fast Application Notification (FAN) is a high availability notification mechanism that Oracle RAC uses to notify other processes about cluster configuration and service-level information, including status changes such as `UP` or `DOWN` events.

One of the main requirements of a highly available application is for it to be quickly notified when something happens to critical system components. Notification enables the connection pool or the application server container to execute event-handling programs. FAN is a timely reaction to cluster resource re-organizations, or resource changes (such as an increase or decrease in the amount of resource).

FAN enables the automated recovery of applications when cluster components fail. For cluster configuration changes, the Oracle RAC high availability framework publishes a FAN event immediately after a change occurs regarding the state of the instances in the cluster. Instead of waiting for the application to query the database and detect a problem, applications can receive FAN events and react immediately.

FAN `UP` and `DOWN` events can apply to instances, services, nodes, and public networks. FAN also publishes Load Balancing Advisory events. FAN `UP` and `DOWN` events provide the following benefits:

- For `DOWN` events, the disruption to the application can be minimized because sessions that are connected to the failed instance or node can be terminated. Incomplete transactions can be terminated and the application user notified immediately. Application users who request new connections are directed to instances that are started and are providing the requested service.
- For `UP` events, when services and instances are started, new connections can be created so that the application can immediately take advantage of the extra resources.

Oracle Clusterware and Oracle RAC use Oracle Notification Service (ONS) to propagate FAN messages both within the Oracle cluster and to client or mid-tier machines. ONS is installed with Oracle Clusterware and the resources to manage the ONS daemons are created automatically during the installation process. ONS daemons run on each node of the cluster and send and receive messages from a configured list of nodes where other ONS daemons are active; this list of nodes can include nodes outside the cluster, such as application server tiers or client nodes. Auto-configuration of Oracle clients, where the client discovers the ONS daemons using their TNS connect string or URL, was introduced with Oracle Database 12*c*.

FAN can also be used with Oracle Data Guard, Active Data Guard, Oracle WebLogic Server Active GridLink for RAC, Universal Connection Pool (UCP) clients, Global Data Services, and OCI-based clients including OCI/OCCI, ODP.NET and OCI Session Pool, among others.

> **See Also:**
>
> - "About Connection Load Balancing"
> - "About the Load Balancing Advisory"

8.1.5 About FAN Callouts

FAN callouts are server-side executable files that Oracle RAC runs immediately when high availability events occur.

A callout is essentially a shell script or precompiled executable written in any programming language. Some examples of how you can use FAN callouts to automate the actions performed when events occur in a cluster configuration are as follows:

- Starting and stopping server-side applications
- Relocating low-priority services when high-priority services come online
- Sending text or numeric messages to pagers

The executable files for FAN callouts are stored in the `Grid_home/racg/usrco` subdirectory. If this subdirectory does not exist in your Grid home, then you must create this directory with the same permissions and ownership as the `Grid_home/racg/tmp` subdirectory.

All executables in the `Grid_home/racg/usrco` subdirectory are executed immediately, in an asynchronous fashion, when a FAN event is received through the Oracle Notification Service (ONS). For most event types, the callout is executed on one node in the cluster (the node generating the event), thus a copy of the executable files used by FAN callouts should be available on every node that runs Oracle Clusterware. An example of a callout script can be found in the section "Appendix D Sample Callout program (PERL based)" in the following white paper, available on Oracle Technology Network: http://www.oracle.com/technetwork/database/options/clustering/overview/fastapplicationnotification12c-2980342.pdf

See Also:

- *Oracle Database Administrator’s Guide* for more information about using FAN callouts
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about configuring Fast Application Notification and FAN callouts

8.1.6 About Application Continuity for Masking Outages

Application Continuity can mask database outages for both planned and unplanned events.

Following planned and unplanned outages, Application Continuity attempts to mask the outage by rebuilding the database session, and resubmitting the in-flight work following recoverable errors that make the database session unavailable. When Application Continuity is configured, an end-user request is replayed if it did not complete provided that replay was enabled when the outage occurs and replay starts within configurable limits (the `replay_init_time` attribute of the service). When a component fails or becomes unresponsive, Application Continuity attempts to restore the database sessions to the time of the instance failure. When replay succeeds, this feature masks applications from transient outages (such as session failure, instance or node outages, network failures and so on) and from planned outages such as repairs, configuration changes and patching the software.

For planned database server outages, the connection pool stops dispensing connections from the instance scheduled for maintenance. New connection requests are routed to the surviving instances where the service is available. On receipt of the Fast Application Notification (FAN) event by the Oracle connection pool or Oracle JDBC driver, idle connections to the down service are removed, and no further connections to that instance are allowed.

- Preparing to Use Application Continuity
  Before you can use Application Continuity, you must complete various checks.

8.1.6.1 Preparing to Use Application Continuity

Before you can use Application Continuity, you must complete various checks.

Application Continuity works on the assumption that the applications are well written in terms of connection usage: borrowing connections and returning them to the connection pool instead of pinning connections.

Before you can use Application Continuity, you must configure attributes for the database services used by the applications. The beginning and end request boundary needs to be added if one of the integrated pools (UCP, WLS data source) is not being used. You may also need to mark request boundaries if not returning connections to the Oracle Connection Pool and a property is not available to unpin.

To use Application Continuity with an Oracle RAC database, use the following configuration checklist:

Service Configuration Checks

- Create a database service (also referred to as an *application service*) to connect to the database. Do not use the Oracle SID, instance name, or the default database service to connect to an Oracle RAC or Oracle Data Guard database.
- For the service, set `failovertype` to `TRANSACTION`, `commit_outcome` to `TRUE` and `notification` to `TRUE`. Optionally, to find the best connections to use, set `rlb_goa` to `SERVICE_TIME` and `clb_gloal` to `SHORT`. See "Creating Services Using SRVCTL".

Software Configuration Checks (Database and Middle-tier)

- Use Oracle Database 12*c* Release 1 (12.1) or later.
- Use any of the following:
  - ODP.NET, Unmanaged Driver
  - OCI Session Pool
  - Universal Connection Pool
  - WebLogic Server
  - JDBC Thin Oracle replay driver
  - Oracle Tuxedo
  - SQL*Plus
- If a statement cache at the application server level is enabled (for example, the WebLogic or third-party application server statement cache), then this *must* be disabled when the replay is used. Instead, configure the JDBC statement cache, which supports Application Continuity. The JDBC statement cache also performs better because it is optimized for JDBC and Oracle. Use `oracle.jdbc.implicitstatementcachesize=`*nnn*.
- Use FAN and FCF for WebLogic Active GridLink Data Source, Universal Connection Pool, ODP.NET Connection Pool or OCI driver or, as an option, use JDBC 12.2 driver that contains FAN for third-party Java containers and applications.
- Check the resource requirements; ensure there is sufficient CPU and memory at the middle tier.

Application Continuity is managed on the server-side. On the client side, additional CPU costs are incurred for garbage collection. The replay driver requires more memory than the base driver because the calls are retained until the end of a request. At the end of the request, the calls are released to the garbage collector. This action differs from the base driver that releases closed calls.

> **Note:**
>
> CPU overhead is reduced on the database side for platforms with current Intel and SPARC chips where validation is assisted by the hardware.

- Determine whether the application uses **mutables**, or sequential data that might be different on a different instance. If so, then you configure the application to keep the original values for `SYSDATE`, `SYSTIMESTAMP`, `SYS_GUID`, and sequences during failover.

  See *Oracle Database SQL Language Reference* for information on issuing the `GRANT` privilege to users for `KEEP [DATE TIME|SYSGUID|`*`Sequence`*`]`.
- Grant `EXECUTE` on the `DBMS_APP_CONT` PL/SQL package to the users that will use Application Continuity.
- After consulting the application certification, assign the `GRANT` privilege for keeping mutables to the users that will use replay.
- Check with the application developers to determine if any requests should not be replayed. The application must explicitly call an API to disable replay for those requests. Instead of modifying the application you can instead use a connection to a service that is not enabled for Application Continuity for this request.

Application Code Checks (consult with the application developer)

Application Continuity masks outages with few or no application changes when you use the Oracle integrated stack. The Application Continuity for OCI solution is embedded in SQL*Plus, OCI Session Pool, Oracle Tuxedo, Oracle WebLogic Active GridLink, ODP.NET, Unmanaged Driver, and is also available with the OCI driver. Applications must be tested to ensure that they are suitable for replay before releasing with Application Continuity.

For information about ensuring automatic application continuity for an application, refer to *Oracle Real Application Clusters Administration and Deployment Guide*.

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for more information on how Application Continuity works, and how you can use it in applications
- *Oracle Database Concepts* for more information about transactions
- *Oracle Database JDBC Developer’s Guide* for more information about Application Continuity for Java applications
- "Creating Services"
- *Oracle Data Provider for .NET Developer's Guide for Microsoft Windows* for information about installing the ODP.NET Unmanaged Drivers
- *Oracle Call Interface Programmer's Guide* for information about OCI Session Pooling
- *SQL*Plus User's Guide and Reference*
- *Administering JDBC Data Sources for Oracle WebLogic Server* for information about using Active GridLink data sources
- *Administering JDBC Data Sources for Oracle WebLogic Server* for information about configuring Oracle WebLogic Server to use Application Continuity

8.1.7 About the Load Balancing Advisory

The Load Balancing Advisory provides information to applications or clients about the current service levels that the Oracle RAC database instances are providing.

Applications can take advantage of the load balancing Fast Application Notification (FAN) events to direct work requests to the instance in the cluster that provides the best performance based on the workload management directives that you have defined for that service. Also, when an instance is restarted, Oracle RAC uses FAN events to notify the application's connection pool so that the connection pool can create connections to the recently started instance and take advantage of the additional resources that this instance provides.

The load balancing advisory is integrated with the Automatic Workload Repository built into Oracle Database 12*c*. The Automatic Workload Repository measures response time and CPU consumption for each service.

The advice given by the Load Balancing Advisory takes into account the power of the server and the current workload of the service on the server. Enabling the Load Balancing Advisory helps improve the throughput of applications by not sending work to instances that are overworked, running slowly, not responding, or have failed.

Your application can take advantage of the Load Balancing Advisory without any programmatic changes if you use an integrated Oracle client, one that has the Run-time Connection Load Balancing feature. Due to the integration with FAN, Oracle integrated clients are more aware of the current status of an Oracle cluster. This prevents client connections from waiting or trying to connect to an instance that is no longer available. The integrated clients for FAN events include the Universal

Connection Pool (UCP) used by Oracle Database 12*c* JDBC, Oracle Database 12*c* ODP.NET, and Oracle Database 12*c* Oracle Call Interface (OCI).

You configure your Oracle RAC environment to use the Load Balancing Advisory by defining service-level goals for each service used. Defining a service-level goal enables the Load Balancing Advisory for that service and enables the publication of FAN load balancing events. There are two types of service-level goals for Run-time Connection Load Balancing:

- Service Time. The Load Balancing Advisory attempts to direct work requests to instances according to their response time. Load Balancing Advisory data is based on the elapsed time for work done by connections using the service, and the available bandwidth to the service. This goal is best suited for workloads that require varying lengths of time to complete, for example, an internet shopping system.
- Throughput. The Load Balancing Advisory measures the percentage of the total response time that the CPU consumes for the service. This measures the efficiency of an instance, rather than the response time. This goal is best suited for workloads where each work request completes in a similar amount of time, for example, a trading system.

If you do not select the Enable Load Balancing Advisory option, then the service-level goal is set to None, which disables load balancing for that service.

See Also:

- "About Fast Application Notification (FAN)"
- "Configuring Clients for High Availability"
- "Administering Services"
- "About Continuous Service Availability with Database Services"
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about integrated Oracle clients

8.1.8 About Connection Load Balancing

Oracle Net connection load balancing spreads user connections across all the instances that support the service used to connect to the database.

Oracle Net is a software component that resides on the client and on the Oracle database server. It establishes and maintains the connection between the client application and the server, and exchanges messages between them using industry standard protocols. For the client application and a database to communicate, the client application must specify location details for the database it wants to connect to, and the database must provide some sort of identification or address.

On the database server, the Oracle Net Listener, commonly known as the listener, is a process that listens for client connection requests. The configuration file for the listener is the `listener.ora`.

Oracle Database 12*c* database clients use SCAN to connect to the database. SCAN can resolve to multiple IP addresses, reflecting multiple listeners in the cluster handling public client connections. Oracle recommends against using the easy connect method with SCAN as it provides no high availability support. Instead, applications should use an Oracle Net connect descriptor with the following format:

```
(DESCRIPTION =
      (CONNECT_TIMEOUT=90) (RETRY_COUNT=20)(RETRY_DELAY=3)
(TRANSPORT_CONNECT_TIMEOUT=3)
       (ADDRESS_LIST =
             (LOAD_BALANCE=on)
             (ADDRESS = (PROTOCOL = TCP)(HOST=primary-scan)(PORT=1521))
             (CONNECT_DATA=(SERVICE_NAME = service_name))))
```

SCAN represents the SCAN for your cluster. If you do not specify a port number, then the default value of 1521 is used for the TCP port identifier. The *service_name* is the name of a database service.

You can also use Net Configuration Assistant (NETCA) to create a **net service name**. The net service name resolves to the connect descriptor. The address portion of the connect descriptor is actually the protocol address of the listener. The client uses a connect descriptor to specify the database or instance to which the client wants to connect.

When a net service name is used, establishing a connection to a database instance takes place by first mapping the net service name to the connect descriptor. This mapped information is stored in one or more repositories of information that are accessed using naming methods. The most commonly used naming method is Local Naming, where the net service names and their connect descriptors are stored in a localized configuration file named `tnsnames.ora`.

When the client connects to the cluster database using a service, you can use the Oracle Net connection load balancing feature to spread user connections across all the instances that are supporting that service. There are two types of load balancing that you can implement: client-side and server-side load balancing. In an Oracle RAC database, client connections should use both types of connection load balancing. When you create an Oracle RAC database using Oracle Database Configuration Assistant (DBCA), DBCA configures and enables server-side load balancing by default.

- About Client-Side Load Balancing
  Client-side load balancing balances the connection requests across the listeners.
- About Server-Side Load Balancing
  With server-side load balancing, the listener directs a connection request to the best instance currently providing the service by using information from the Load Balancing Advisory.

See Also:

- "About Continuous Service Availability with Database Services"
- "Verifying Oracle Net Supports Newly Created Services"
- *Oracle Database 2 Day DBA* for more information about network configurations
- *Oracle Database 2 Day DBA* for more information about client connections
- *Oracle Database 2 Day DBA* for more information about the tools you can use for configuring the network

8.1.8.1 About Client-Side Load Balancing

Client-side load balancing balances the connection requests across the listeners.

When the listener receives the connection request, the listener connects the user to an instance that the listener knows provides the requested service.

Client-side load balancing is defined in your client connection definition by setting the parameter `LOAD_BALANCE=yes` in the `tnsnames.ora` file. When you set this parameter to `yes`, the Oracle client randomly selects an address from the address list, and connects to that node's listener. This balances client connections across the available listeners in the cluster.

When you create an Oracle RAC database using DBCA, the assistant creates a sample load balancing connection definition in the `tnsnames.ora` file.

Client-side load balancing includes connection failover. With connection failover, if an error is returned from the chosen address, then Oracle Net Services tries the next address in the address list until either a successful connection is made or it has exhausted all the addresses in the list.

See Also:

- "About Connection Load Balancing"
- "About the Load Balancing Advisory"
- *Oracle Database 2 Day DBA* for more information about network configurations
- *Oracle Database 2 Day DBA* for more information about client connections
- *Oracle Database 2 Day DBA* for more information about the tools you can use for configuring the network

8.1.8.2 About Server-Side Load Balancing

With server-side load balancing, the listener directs a connection request to the best instance currently providing the service by using information from the Load Balancing Advisory.

For each service, you can define the method the listener uses for load balancing by setting the connection load balancing goal. You can use a goal of either long or short for connection load balancing. These goals have the following characteristics:

- Short—When the connection load balancing goal is set to `SHORT`, the connections are distributed based on response time or throughput depending on which Run-time Connection Load Balancing goal is set. Use the Short connection load balancing goal for applications that have connections of brief duration.
- Long—Connections are distributed across instances based on the number of sessions in each instance, for each instance that supports the service. Use the Long connection load balancing goal for applications that have connections of long duration. This is typical for connection pools and SQL*Forms sessions. Long is the default connection load balancing goal.

> **Note:**
>
> If you did not use DBCA to create your database, or if you are using listener ports other than the default of 1521, then you must configure the `LOCAL_LISTENER` and `REMOTE_LISTENER` database initialization parameters for your cluster database to point to *`SCAN:port`*.

> **See Also:**
>
> - "About Connection Load Balancing"
> - "About the Load Balancing Advisory"

8.1.9 About Run-time Connection Load Balancing

Run-time Connection Load Balancing is a feature of Oracle connection pools that can distribute client work requests across the instances in an Oracle RAC database based on the Load Balancing Advisory information.

The connection allocation is based on the current performance level provided by the database instances as indicated by the Load Balancing Advisory FAN events. This provides load balancing at the transaction level, instead of load balancing at the time of the initial database connection.

With Run-time Connection Load Balancing, applications use Load Balancing Advisory information to provide better performance to users. OCI Session pools and ODP.NET connection pools support Run-time Connection Load Balancing. For Java applications, Oracle recommends the Universal Connection Pool (UCP). The Universal Connection

Pool is integrated to take advantage of Load Balancing Advisory information. UCP, introduced in Oracle Database 11*g* patch set 1 (11.1.0.7), can be used against Oracle Database 10*g*, Oracle Database 11*g*, or Oracle Database 12*c*.

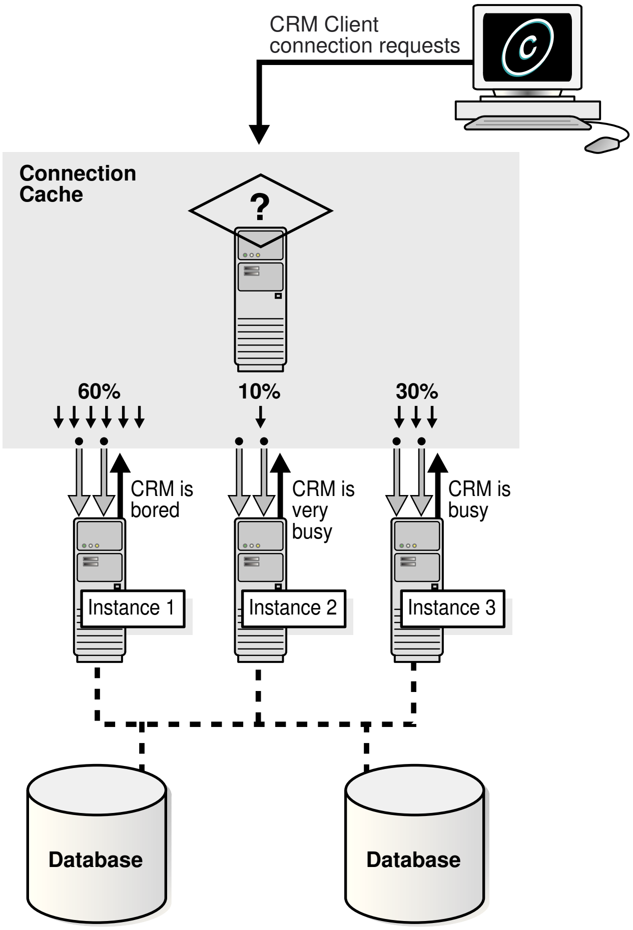
You must enable the client data source for Run-time Connection Load Balancing with a service that has the following configuration:

- The Load Balancing Advisory is enabled and the service-level goal is set to `SERVICE_TIME`, `THROUGHPUT` , or `NONE`.
- The service connection load balancing goal is set to Short.

The illustration in the steps below illustrates Run-time Connection Load Balancing. In this illustration, the Oracle RAC database has three instances. Suppose that the Load Balancing Advisory indicates that Instance1 and Instance3 have the best performance, while Instance2 currently has less than optimal performance. When Run-time Connection Load Balancing is enabled for the Universal Connection Pool, the following process occurs:

1. A client requests a connection from the connection pool.
2. Run-time Connection Load Balancing selects the connection that belongs to the most efficient (best) instance from the connection pool. In the illustration in the next step, there are three possible nodes to which the connection can be routed. Instance1, which has the least amount of CPU workload, is currently being assigned about 60 percent of the incoming connections. Instance2, which is currently overloaded, is only being assigned around 10 percent of the incoming connections. Instance3, which has a high workload, is being assigned around 30 percent of the incoming connections. The best instance to handle the connection request in this case would be Instance1.
3. The client receives the connection that would process the work request with the best response time.

Figure 8-1   Run-time Connection Load Balancing



Starting with Oracle Database 11*g* you can use an additional flag in the load balancing advisory event called **affinity hint**. The affinity hint is a flag that indicates if affinity is a benefit for a particular instance and service combination. Different instances offering the same service can have different settings for the affinity hint.

The affinity hint is automatic when the load balancing advisory is turned on through setting the goal on the service. This flag is for temporary affinity that lasts for the duration of a web session. Web conversations often connect and disconnect many times during the entire session. During each of these connects, it may access the same or similar data, for example, a shopping cart, Siebel, and so on. Affinity can improve buffer cache efficiency, which lowers CPU usage and transaction latency.

Applications using Oracle Database 12*c* and UCP can take advantage of this new affinity feature. If the affinity flag is turned on in the Load Balancing Advisory event, then UCP creates an Affinity Context for the web session such that when that session does a get connection from the pool, the pool always tries to give it a connection to the instance it connected to the first time it acquired a session. The choice of instance for the first connection is based on the current load balancing advisory information.

Related Topics

- About Connection Load Balancing
  Oracle Net connection load balancing spreads user connections across all the instances that support the service used to connect to the database.
- About the Load Balancing Advisory
  The Load Balancing Advisory provides information to applications or clients about the current service levels that the Oracle RAC database instances are providing.
- About FAN Callouts
  FAN callouts are server-side executable files that Oracle RAC runs immediately when high availability events occur.

8.2 Creating Services

You can create a service using Oracle Enterprise Manager or the SRVCTL utility.

> **Note:**
>
> The `DBMS_SERVICE` package will create the database structures necessary to offer a service, but none of the Oracle Clusterware functionality is configured, such as service placement, failure characteristics and so on.

To manage workloads, you can define database services that you assign to a particular application or to a subset of an application's operations. You can also use services to manage the workload for different types of work.

- Creating Services Using Enterprise Manager
  You can use either Oracle Enterprise Manager Cloud Control or Oracle Enterprise Manager Database Express to create services.
- Creating Services Using SRVCTL

8.2.1 Creating Services Using Enterprise Manager

You can use either Oracle Enterprise Manager Cloud Control or Oracle Enterprise Manager Database Express to create services.

To create a service using Enterprise Manager:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Availability** menu, select **Cluster Managed Database Services**. Enter or confirm the credentials for the Oracle RAC database and host operating system and click **Continue**.

   The Cluster Managed Database Services page appears.

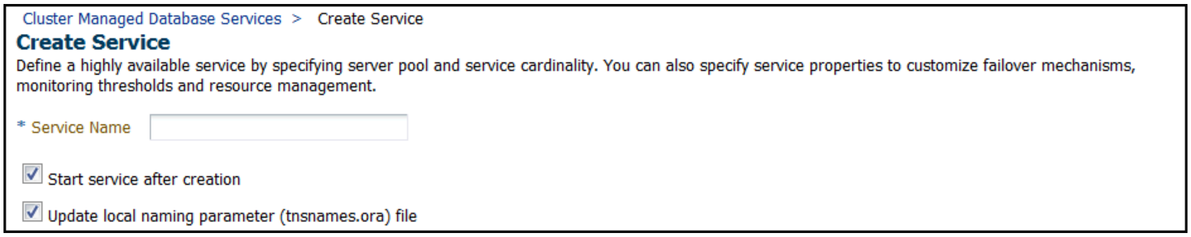
3. Click **Create Service**.

   The Create Service page appears.

4. Enter the name of your service in the Service Name field, for example, `DEVUSERS`.

5. Select **Start Service after creation** if you want the service to be started after it is created. Select **Update local naming parameter (tnsnames.ora) file** if you want Oracle Enterprise Manager Cloud Control to add the new service to the local Oracle Net Services `tnsnames.ora` file.

Figure 8-2   Create Service Page of Oracle Enterprise Manager



Cluster Managed Database Services > Create Service

Create Service

Define a highly available service by specifying server pool and service cardinality. You can also specify service properties to customize failover mechanisms, monitoring thresholds and resource management.

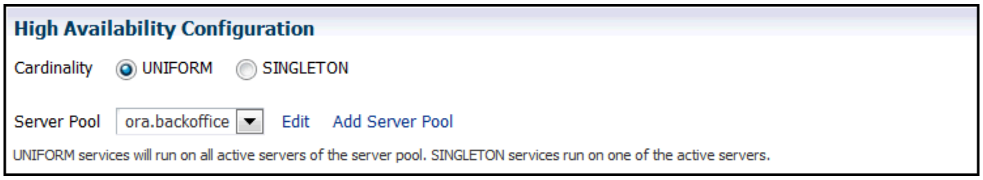
* Service Name

☑ Start service after creation

☑ Update local naming parameter (tnsnames.ora) file

6. (Policy-managed databases only) For the service type, select **UNIFORM** or SINGLETON.

Figure 8-3   High Availability Configuration Section on the Create Service Page



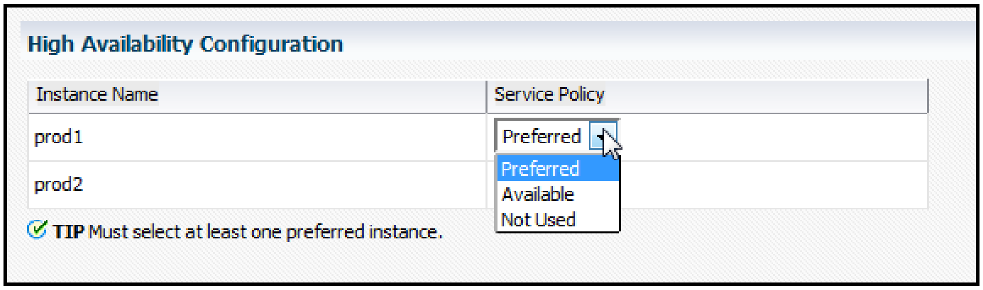
High Availability Configuration

Cardinality ● UNIFORM ○ SINGLETON

Server Pool ora.backoffice Edit Add Server Pool

UNIFORM services will run on all active servers of the server pool. SINGLETON services run on one of the active servers.

7. (Administrator-managed databases only) For each instance choose whether that instance is a Preferred instance or an Available instance. If you do not want the service to run on an instance under any circumstances, then set the Service Policy for that instance to Not Used.



High Availability Configuration

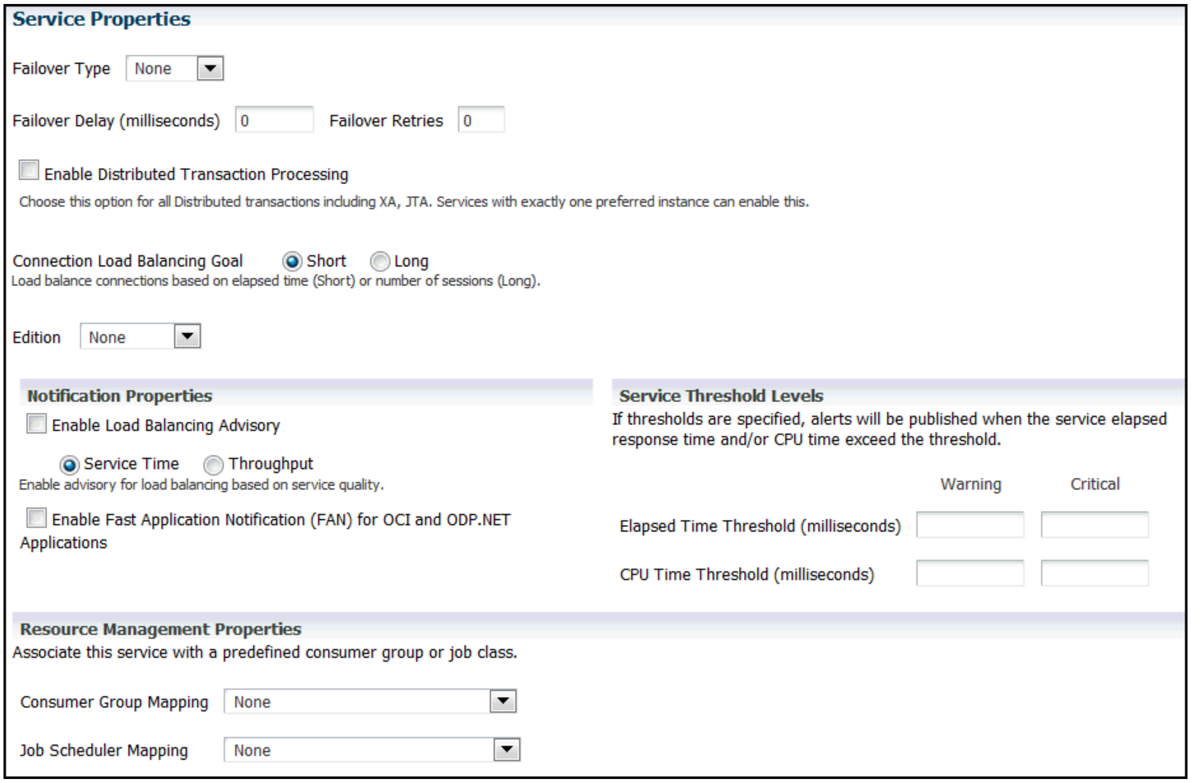
Instance Name	Service Policy
prod1	Preferred
prod2	

✓ TIP Must select at least one preferred instance.

8. In the Service Properties section, select **Short** for Connection Load Balancing Goal to distribute the connection workload based on elapsed time instead of the overall number of connections. Otherwise, choose **Long**.

9. Select **Enable Load Balancing Advisory** under the sub-heading Notification Properties to enable the Load Balancing Advisory for this service, as shown in the following screenshot. Choose a service-level goal of either **Service Time** or **Throughput**.

Figure 8-4 Bottom Portion of the Create Service Page



Service Properties

Failover Type: None ▾

Failover Delay (milliseconds): 0 Failover Retries: 0

□ Enable Distributed Transaction Processing

Choose this option for all Distributed transactions including XA, JTA. Services with exactly one preferred instance can enable this.

Connection Load Balancing Goal ● Short ○ Long

Load balance connections based on elapsed time (Short) or number of sessions (Long).

Edition: None ▾

Notification Properties

□ Enable Load Balancing Advisory

● Service Time ○ Throughput

Enable advisory for load balancing based on service quality.

□ Enable Fast Application Notification (FAN) for OCI and ODP.NET Applications

Service Threshold Levels

If thresholds are specified, alerts will be published when the service elapsed response time and/or CPU time exceed the threshold.

	Warning	Critical
Elapsed Time Threshold (milliseconds)		
CPU Time Threshold (milliseconds)		

Resource Management Properties

Associate this service with a predefined consumer group or job class.

Consumer Group Mapping: None ▾

Job Scheduler Mapping: None ▾

10. Select **Enable Fast Application Notification** under the heading Notification Properties if an Oracle Call Interface (OCI) or ODP.NET application uses this service, and you want to enable FAN.

11. In the Service Threshold Levels section, you can optionally set the service-level thresholds by entering a value in milliseconds for Warning and Critical thresholds for the Elapsed Time and CPU Time metrics.

12. If you want to use a Resource Plan to control the resources used by this service, then select the name of the consumer group from the Consumer Group Mapping list in the Resource Management Properties section. For example, you might choose the `LOW_GROUP` consumer group to give development users low priority to database resources.

> **Note:**
>
> You cannot change the consumer group name for a service on the Edit Service page. This is because there may be several consumer groups associated with a given service. However, the Edit Service page contains a link to the Resource Consumer Group Mapping page, where you can modify the consumer group mapping for the service.

13. If a specific Oracle Scheduler job class uses this service, then you can specify the mapping by selecting the name from the Job Scheduler Mapping list in the Resource Management Properties.

14. Click **OK** to create the service.

See Also:

- "About Connection Load Balancing"
- "About the Load Balancing Advisory"
- "About Fast Application Notification (FAN)"
- "Administering Services"
- *Oracle Database Administrator's Guide* for more information about the Oracle Scheduler

8.2.2 Creating Services Using SRVCTL

SRVCTL is a command-line interface that you can use to create services and administer Oracle Database objects that Oracle Clusterware manages. For functionality that is not available with Oracle Enterprise Manager Database Express, you can use SRVCTL commands instead. See *Oracle Real Application Clusters Administration and Deployment Guide* for information about how to use the SRVCTL utility.

To configure services for Application Continuity or Transaction Guard using SRVCTL:

1. To configure services for Application Continuity, you must set the service attributes `failovertype` to `TRANSACTION`, `commit_outcome` to `TRUE`, and `notification` to `TRUE`. Additionally, you can set values for these other service attributes for Application Continuity and load balancing:

   - `replay_init_time`: Specifies the time in seconds after which replay will not be started; example settings are 60, 300, or 600 seconds; the recommended value is 900 seconds.
   - `retention`: Specifies the time (in seconds) that the commit outcome is retained for Transaction Guard. Recommended value is 86400 (24 hours).
   - `failoverretry`: Number of connection retries for each replay attempt; recommended value is 30.
   - `failoverdelay`: Delay in seconds between connection retries; recommended value is 10.
   - `clbgoal`: For connection load balancing, use `SHORT` when using run-time load balancing.
   - `rlbgoal`: Set to `SERVICE_TIME`.
   - `session_state`: For Application Continuity support for application requests with more than one database transaction; recommended value is `DYNAMIC`.
   - `drain_timeout`: The number of seconds allowed for resource draining to be completed. Permitted values are "", 0, or any positive integer. The default value is an empty string, which means that this option is not set. If it is zero, then draining occurs immediately.

To use SRVCTL to modify the service attributes, use a command similar to the following, where *racdb* is the name of your Oracle RAC database, and *app1* is the name of the service you are modifying:

```
srvctl modify service -db racdb -service app1 -clbgoal SHORT
-rlbgoal SERVICE_TIME -failoverretry 30 -failoverdelay 10
-failovertype TRANSACTION -replay_init_time 1800 -retention 86400
-notification TRUE commit_outcome TRUE
```

2. To create a service for Application Continuity for a policy-managed Oracle RAC database, use a command similar to the following, where *racdb* is the name of your Oracle RAC database, *app2* is the name of the service you are modifying, and *Svrpool1* is the name of the server pool in which the service is offered:

```
srvctl create service -db racdb -service app2 -serverpool ora.Srvpool1
-clbgoal SHORT -rlbgoal SERVICE_TIME -failoverretry 30
-failoverdelay 10 -failovertype TRANSACTION -replay_init_time 1800
-retention 86400 -notification TRUE -commit_outcome TRUE
```

8.3 Administering Services

You can create and administer services using Enterprise Manager. You can also use the SRVCTL utility to perform most service management tasks.

- About Service Administration Using Enterprise Manager
  The Cluster Managed Database Services page is the master page for beginning all tasks related to services.
- Using the Cluster Managed Database Services Page
  When managing services using Enterprise Manager, you use the Cluster Managed Database Services page.
- Verifying Oracle Net Supports Newly Created Services
  By configuring and using a SCAN for your cluster, you no longer have to modify the network setup on each client. Using the `lsnrctl` utility you can verify that the new service is recognized by Oracle Net and available to the database clients.
- Managing Groups of Services
  Using SRVCTL, you can manage groups of services with a single command.

> **See Also:**
>
> - "About Oracle Database Services"
> - "Creating Services"

8.3.1 About Service Administration Using Enterprise Manager

The Cluster Managed Database Services page is the master page for beginning all tasks related to services.

To access this page, go to the Cluster Database Maintenance page, then click **Cluster Managed Database Services** in the Services section. You can use this page and links from this page to do the following:

- View a list of services for the cluster.
- View the instances on which each service is currently running.
- View the status for each service.
- Create or edit a service.
- Start or stop a service.
- Enable or disable a service.
- Perform instance-level tasks for a service.
- Delete a service.

> **See Also:**
>
> - "About Oracle Database Services"
> - "Creating Services"

8.3.2 Using the Cluster Managed Database Services Page

When managing services using Enterprise Manager, you use the Cluster Managed Database Services page.

To access the Cluster Managed Database Services page:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.
2. From the **Availability** menu, select **Cluster Managed Database Services**. Enter or confirm the credentials for the Oracle RAC database and host operating system and click **Continue**.

   The Cluster Managed Database Services page appears and displays the services that are available on the cluster database instances.
3. Perform any of the following tasks using this page:
   - View a list of services for the cluster, the instances on which each service is currently running, the status of each service, the server pool associated with the service, the responsiveness of the service, and any alerts or messages for that service.
   - Start or stop a service.
   - Test the connection for a service, or display the Oracle Net TNS strings being used.
   - Access the Create Service page.
   - Edit, delete, enable, or disable a service.
   - Access the Edit Server Pool page for the server pool associate with each service.

See Also:

- "About Service Administration Using Enterprise Manager"
- "About Oracle Database Services"
- "Creating Services"

8.3.3 Verifying Oracle Net Supports Newly Created Services

By configuring and using a SCAN for your cluster, you no longer have to modify the network setup on each client. Using the `lsnrctl` utility you can verify that the new service is recognized by Oracle Net and available to the database clients.

Note:

When using utilities to manage your cluster, databases, database instances, Oracle Automatic Storage Management (Oracle ASM), and listeners, use the appropriate binary that is in the home directory of the object or component you are managing. Also, set your `ORACLE_HOME` environment variable to point to this directory.

For example, to use `lsnrctl` to manage the SCAN listener, use the binaries located in the Grid home (where the listener is running), and set the `ORACLE_HOME` environment variable to the location of the Grid home.

To verify Oracle Net Services supports the newly created service:

1. Determine if the listener on the local node recognizes the new service by using the following command:

   ```
   lsnrctl status LISTENER_SCAN1
   ```

   You should see a list for the new service, similar to the following:

   ```
   Services Summary...
   Service "DEVUSERS.example.com" has 2 instance(s).
     Instance "sales1", status READY, has 1 handler(s) for this service...
     Instance "sales2", status READY, has 1 handler(s) for this service...
   ```

   The displayed name for your newly created service, for example `DEVUSERS.example.com`, is the service name you use in your connection strings.

2. (Optional) Use a text editor to modify the `tnsnames.ora` file in the *`Oracle_home`*`/network/admin` directory on each node that contains an instance listed as a Preferred or Available instance for the service, and on each client that uses the service to connect to the database. Add an entry similar to the following, specifying the VIP address for each node:

   ```
   DEVUSERS =
     (DESCRIPTION =
       (ADDRESS_LIST =
         (ADDRESS = (PROTOCOL = TCP)(HOST = docrac1-vip)(PORT = 1521))
   ```

```
      (ADDRESS = (PROTOCOL = TCP)(HOST = docrac2-vip)(PORT = 1521))
    (LOAD_BALANCE = yes)
    )
  (CONNECT_DATA = (SERVICE_NAME = DEVUSERS.oracle.com))
   )
```

In the previous example, the `ADDRESS_LIST` parameter contains one `ADDRESS` for each node that contains an instance configured as either Preferred or Available for the service.

3. Test the Oracle Net Services configuration by attempting to connect to the Oracle RAC database using SQL*Plus and the SCAN.

   The connect identifier for easy connect naming has the following format:

   ```
   "[//]SCAN[:port]/service_name"
   ```

   SCAN is the SCAN for your cluster, which defaults to *cluster_name*.*GNS_sub_domain*. The *service_name* is the name of the database service you want to use for connecting to the database. You can optionally specify the TCP port number on which the Oracle SCAN listener listens for connections.

   For example, you might use the following commands to connect to the DEVUSERS service for Oracle RAC database in the `docrac` cluster, which belongs to the `example.com` domain:

   ```
   $ sqlplus system@\"docrac.example.com/DEVUSERS.example.com\"
   Enter password: password
   ```

   After you enter the password, you should see a message indicating you are successfully connected to the Oracle RAC database. If you get an error message, then examine the connect identifier and verify the user name, password, and service name were typed in correctly and all the information is correct for your Oracle RAC environment.

See Also:

- "About Connection Load Balancing"
- *Oracle Database 2 Day DBA* for details on how to view the listener configuration
- *Oracle Database 2 Day DBA* for details on how to connect to an Oracle database from a client computer

8.3.4 Managing Groups of Services

Using SRVCTL, you can manage groups of services with a single command.

Many enterprises run a large number of services, whether it be many services offered by a single database or instance, or many databases offering a few services running on the same node. You no longer need to run SRVCTL commands for each individual service but need only specify the node name, database name, pluggable database (PDB) name, or the instance name for all affected services.

Example 8-1 Starting All Services on a Node

To start all services running on a given node, use a command similar to the following:

```
srvctl start service –node racnode01
```

Example 8-2 Starting All Services for a Database

To start all services supported by a database, use a command similar to the following:

```
srvctl start service –db racdb
```

To start all services for a database instance, use a command similar to the following:

```
srvctl start service –db racdb –inst racdb1
```

To start all services for a pluggable database, use a command similar to the following:

```
srvctl start service –db racdb -pdb pdb_region1
```

Example 8-3 Stopping All Services on a Node

To stop all services running on the specified node, giving connections 60 seconds to migrate to a surviving instance before being disconnected, use a command similar to the following:

```
srvctl stop service –node racnode01 -drain_timeout 60 -stopoption IMMEDIATE
```

This command stops all services running on `racnode01`, allowing a drain interval of 60 seconds. After 60 seconds any remaining sessions will be stopped using the IMMEDIATE option. The 60 second drain timeout setting overrides any attribute setting on a given service.

Example 8-4 Stopping All Services for a Database

To stop all services supported by a database and any of its PDBs, use a command similar to the following:

```
srvctl stop service –db racdb –drain_timeout 15 –stopoption IMMEDIATE
```

This command stops all services for the `racdb` database, allowing a drain interval of 15 seconds. After 15 seconds any remaining sessions will be stopped using the IMMEDIATE option. The 15 second drain timeout setting overrides any attribute setting on a given service.

To stop all services for a database instance, use a command similar to the following:

```
srvctl stop service –db racdb –inst racdb1 –drain_timeout 15 –stopoption IMMEDIATE
```

This command stops all services for the `racdb1` database instance, allowing a drain interval of 15 seconds. After 15 seconds any remaining sessions will be stopped using the IMMEDIATE option. The 15 second drain timeout setting overrides any attribute setting on a given service.

To stop all services for a pluggable database, use a command similar to the following:

```
srvctl stop service –db racdb –pdb pdb_region1 –drain_timeout 15 –stopoption
TRANSACTIONAL
```

This command stops all services for the `pdb_region1` pluggable database, allowing a drain interval of 15 seconds. After 15 seconds any remaining sessions will be stopped using the TRANSACTIONAL option. The 15 second drain timeout setting overrides any attribute setting on a given service.

You can also specify a list of services to stop for a specific database. For example, to stop only some of the services offered by a PDB, use a command similar to the following:

```
srvctl stop service –db racdb –pdb pdb_region1 –service “crm_service, hr_service,
batch_service” –drain_timeout 60 –stopoption IMMEDIATE
```

In the previous command, `crm_service`, `hr_service`, and `batch_service` are all the names of services.

Example 8-5 Stopping All Database Instances on a Node

To stop all databases or database instances on a node, use a command similar to the following:

```
srvctl stop database -node racnode01 -drain_timeout 60 –stopoption IMMEDIATE -
failover –force
```

When `-failover` is specified, then the following actions occur:

- All services are relocated, if possible, respecting the `–drain_timeout` and `–stopoption` attributes.
- Any services that cannot be failed over are stopped, using an action that corresponds with the value specified for `–stopoption` .
- Services are stopped so no further sessions are accepted for the instances being shut down.
- After a period of time specified by `–drain_timeout`, or after all sessions using the targeted services have disconnected, the instances are shut down.
- The instances are stopped using the method specified by `–stopoption`.

If `–force` is specified, then the following actions occur:

- The remaining services are stopped respecting the `–drain_timeout` and `–stopoption` values.
- Services are stopped so no further sessions are accepted for the instances being shut down.
- After a period of time specified by `–drain_timeout`, or after all sessions using the targeted services have disconnected, the instances are shut down.
- The instances are stopped using the ABORT option.

Example 8-6 Relocating a Group of Services

You can use the `srvctl relocate service` command to relocate a group of services to a target destination, which can be an instance, a node, a database, or a pluggable database. For example:

```
srvctl relocate service –database racdb –oldinst racdb01 –newinst racdb02
  -drain_timeout 30 -force
```

In the preceding command example, all services are relocated from the database instance `racdb01` to the database instance `racdb02`. The services relocate only if the target can support that service, as defined by the service configuration. Any services that cannot be relocated remain at the original location.

8.4 Managing Planned Maintenance Without User Interruption

Using a combination of features, you can stop database services or instances with little to no impact to connected sessions.

You may have to stop database instances for many reasons, such as upgrading the Oracle software, patching, and replacing hardware. The steps in this topic result in the least impact to users and does not interrupt running applications.

You can assign services to one or more instances in an administrator-managed Oracle RAC database or to server pools in a policy-managed Oracle RAC database. If Oracle RAC detects an outage, then Oracle Clusterware isolates the failed component and recovers the dependent components. For services, if the failed component is an instance, then Oracle Clusterware attempts to maintain the current levels of service availability (cardinality of the service). If the service definition allows for failover and that is necessary to maintain cardinality, then failover occurs.

For planned maintenance, the recommended approach is to drain database connections over a controlled time period from fast application notification (FAN)-enabled Oracle connection pools. The service attributes `-drain_timeout` and `-stopoption` control the drain time period, and also define how to handle sessions that have not completed their work within the specified time period. Requests that complete their work and then check back in to the pool or close their connection, can be directed to a new location that is not affected by the planned maintenance.

Both `-drain_timeout` and `-stopoption` are service attributes that can be defined when creating or modifying the service. The advantage of setting the attributes on the service is that they are set once and then used for all future SRVCTL planned maintenance operations. You can also specify these attributes on the SRVCTL command line to override the current settings.

Using any FAN-aware pool with FAN configured (Oracle Call Interface (OCI), Universal Connection Pool (UCP), Database Resident Connection Pooling (DRCP), WebLogic Server, or Oracle Data Provider for .NET), the steps to manage planned maintenance without user interruption are as follows:

1. Use the SRVCTL `relocate` command to relocate the service from the instance being shut down without disrupting the sessions, or if a uniform service is being used, then stop the service on this instance. If the parameter `-stopoption` is used on the SRVCTL command line, then `-force` must be used when either `srvctl stop service` or `srvctl relocate service` is used. For example:

   ```
   srvctl relocate service –database racdb –service myservice –drain_timeout 60
     –stopoption IMMEDIATE –oldinst racdb1 -force
   ```

2. The stop of the service (through issuing either `srvctl relocate service` or `srvctl stop service`) causes the FAN DOWN event to be sent. FAN-aware pools then clear idle sessions immediately and mark active sessions to be released at the next check-in. The pool waits an amount of time equal to the value of `drain_timeout` in seconds for the check-in to occur. If `drain_timeout` is set to zero or NULL, then there is no delay for draining and the service is stopped immediately with the method specified by `-stopoption`.

   The draining interval can be set for a given service, through the service attribute `drain_timeout` or on the SRVCTL command line via `-drain_timeout`.

Existing connections on other instances remain usable, and new connections can be opened to these instances if required.

3. Applications should close all sessions, but in reality this does not always happen. It is best practice to set a time interval (`-drain_timeout`) during which work can complete. When the draining time out interval has elapsed, the services are stopped
4. You can now stop the database instance without disconnecting user sessions.
5. After performing the maintenance operations, restart the instance and services. Run-time load balancing, if enabled, balances the sessions back to use the restored instance. A FAN `UP` event for the service invokes moving the services back to the preferred instances gradually when using a FAN-aware client.

8.5 Configuring Clients for High Availability

There are three central elements to consider when automating failover for application clients.

- First, clients that are connected at the time of failure must be quickly and automatically notified that a failure has occurred to avoid waiting for TCP/IP network time-outs before attempting to connect to the new database instance (such time-outs range anywhere from eight minutes to two hours, depending on operating system). Oracle RAC configurations use Fast Application Notification (FAN) to notify JDBC clients, OCI clients, and ODP.NET clients. FAN event notifications and callouts enable automatic and fast redirection of clients after a site failure.
- The second central element of client failover, is the redirection of new incoming requests to the new instance after a failure has occurred, which can be implemented using application services. When you create and activate application services on several instances of an Oracle RAC database, then if the service becomes unavailable on one instance, clients connections are redirected to other instances where the service is available. Applications need only provide the name of the database service in their connect strings–there is no need to specify an instance name because, using listener cross registration, SCAN listeners and other remote listeners in the cluster are aware of which instances are currently providing a service when a connection request comes in.
- The third central element is masking the outage from the client and the application. Masking outages of the database session is a complex task for application development and, therefore, errors and timeouts are often exposed to the client. Application Continuity attempts to mask outages from applications by replaying incomplete application requests after unplanned and planned outages.

For applications that are unable to take advantage of Application Continuity, two additional features are available: Transparent Application Failover (TAF) for OCI based applications and Transaction Guard. Transaction Guard enables applications to build their own masking of outages by knowing the outcome of the last in-flight transaction.

This section deals with configuring high availability for application clients.

- Configuring Oracle Net Services Parameters For Highly Available Connections
When using services to connect to a database, there are various parameters you can set to configure high availability.

- Configuring Application Clients for High Availability
  You can configure Oracle Call Interface (OCI), Java Database Connectivity (JDBC), and Oracle Data Provider for .NET (ODP.NET) clients to be highly available and use Application Continuity and Transaction Guard.

> **See Also:**
>
> - "About Fast Application Notification (FAN)"
> - "About Oracle Database Services"
> - *Oracle Real Application Clusters Administration and Deployment Guide* for information about client-side connection configuration

8.5.1 Configuring Oracle Net Services Parameters For Highly Available Connections

When using services to connect to a database, there are various parameters you can set to configure high availability.

- For applications that need to connect to an Oracle Database, use the following Oracle Net Services parameters to minimize connection wait times:
  - `CONNECT_TIMEOUT`: controls the amount of time to wait when attempting to connect to an address
  - `RETRY_COUNT`: controls the number of times the application attempts to connect to the database
  - `RETRY_DELAY`: specifies the delay in seconds between subsequent retries for a connection. This parameter works in conjunction with the `RETRY_COUNT` parameter.
  - `TRANSPORT_CONNECT_TIME`: controls the amount of time allowed for a client to establish a TCP connection to the database

  You can specify these parameters in either the timeout section of the `tnsnames.ora` file or at the `DESCRIPTION` level of a connect string, for example:

```
net_service_name=
 (DESCRIPTION=
  (CONNECT_TIMEOUT=60)(RETRY_COUNT=3)(RETRY_DELAY=2)(TRANSPORT_CONNECT_TIMEOUT=3
  (ADDRESS_LIST=
    (LOAD_BALANCE=ON)
    (ADDRESS=(PROTOCOL=tcp)(HOST=docrac-scan.example.com)(PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=racdb.example.com)))
```

Note:

- The `CONNECT_TIMEOUT` parameter is equivalent to the `sqlnet.ora` parameter `sqlnet.OUTBOUND_CONNECT_TIMEOUT` and overrides it.
- You should not configure the `sqlnet.outbound_connection_timeout` parameter on the server `sqlnet.ora` file, as this can impact Oracle Clusterware operations.
- The `TRANSPORT_CONNECT_TIMEOUT` parameter is equivalent to the `sqlnet.ora` parameter `tcp.CONNECT_TIMEOUT` and overrides it.

See Also:

Oracle Database Net Services Reference for more information about the timeout and retry parameters for Oracle Net Services

8.5.2 Configuring Application Clients for High Availability

You can configure Oracle Call Interface (OCI), Java Database Connectivity (JDBC), and Oracle Data Provider for .NET (ODP.NET) clients to be highly available and use Application Continuity and Transaction Guard.

Configuring JDBC Client for High Availability

An application that uses JDBC connections should use a connection pool, so an existing connection can be used in the pool instead of making a new connection to the database every time you access data. The Universal Connection Pool (UCP) is a Java-based connection pool that supports any type of connection (JDBC, LDAP, JCA), to any type of database (Oracle or non-Oracle) with any middle tier (Oracle or non-Oracle). It also supports standalone deployments such as TopLink or BPEL. UCP also supports integration features of Oracle Database such as Fast Connection Failover, Run-time Connection Load Balancing, Web Affinity, Application Continuity, and Transaction Guard with Oracle RAC when using the JDBC Thin driver.

For information on configuring JDBC Clients for high availability, refer to the following:

- "Oracle RAC Fast Application Notification" in *Oracle Database JDBC Developer's Guide* for information about configuring FAN and Oracle Notification System (ONS)
- "About Configuring Oracle JDBC for Application Continuity for Java" in *Oracle Database JDBC Developer's Guide* for information about configuring Application Continuity
- "Ensuring Application Continuity" in *Oracle Universal Connection Pool Developer's Guide* for information about using Application Continuity with the Universal Connection Pool
- ""Transaction Guard for Java" in *Oracle Database JDBC Developer's Guide* for information about configuring Transaction Guard

Configuring OCI Clients for High Availability

The Oracle Call Interface (OCI) provides integration with FAN and Load Balancing Advisory events. To take advantage of the Load Balancing Advisory, you must enable the OCI Session pool.

For information on configuring OCI Clients for high availability, refer to the following:

- "OCI Programming Advanced Topics" in *Oracle Call Interface Programmer's Guide* for information about configuring FAN, Transparent Application Failover, and Runtime Connection Load Balancing
- ""OCI and Transaction Guard" in *Oracle Call Interface Programmer's Guide* for information about configuring Transaction Guard
- ""OCI and Application Continuity" in *Oracle Call Interface Programmer's Guide* for information about configuring Application Continuity

Configuring ODP.NET Clients for High Availability

Oracle Data Provider for .NET (ODP.NET) connection pools subscribe to FAN notifications from Oracle RAC that indicate when nodes are down and when services are up or down. Based on these notifications, the ODP.NET connection pool:

- Interrupts active sessions so they do not wait on TCP/IP timeouts, and
- Clears idle sessions previously connected to the down node or instance or network so they are not passed to user sessions
- Creates new connections to healthy nodes, if possible

ODP.NET provides Run-time Connection Load Balancing to provide enhanced load balancing of the application workload. Instead of randomly selecting an available connection from the connection pool, it chooses the connection that can provide the best service based on the current workload information.

For information on configuring ODP.NET Clients for high availability, refer to the following:

- "Real Application Clusters and Global Data Services" in *Oracle Data Provider for .NET Developer's Guide for Microsoft Windows* for information about configuring FAN, Fast Connection Failover and Runtime Connection Load Balancing
- "Using Transaction Guard to Prevent Logical Corruption" in *Oracle Data Provider for .NET Developer's Guide for Microsoft Windows*
- "GetLogicalTransactionStatus" in *Oracle Data Provider for .NET Developer's Guide for Microsoft Windows* for an example of using Transaction Guard in an ODP.NET application

9 Administering Backup and Recovery

To protect the database against data loss and reconstruct the database after data loss, you must devise, implement, and manage a backup and recovery strategy for your Oracle RAC Database.

- Overview of Oracle RAC Database Backup and Recovery
  To protect your Oracle Real Application Clusters (Oracle RAC) database from hardware failures or disasters, you must have a physical copy of the database files.
- About the Fast Recovery Area in Oracle RAC
  Using a fast recovery area minimizes the need to manually manage disk space for your backup-related files and balance the use of space among the different types of files.
- Archiving the Oracle Real Application Clusters Database Redo Logs
  To make your data highly available, it is important to configure the database so you can recover your data after a system failure. Redo logs contain a record of changes that were made to datafiles.
- About Preparing for Backup and Recovery Operations
  Before taking backups of your Oracle Real Application Clusters (Oracle RAC) database using Enterprise Manager, you must configure access for the user performing the backups, or **credentials**.
- Performing Backups of Your Oracle Real Application Clusters Database
  You can use Oracle Enterprise Manager to backup your Oracle Real Application Clusters (Oracle RAC) database.
- Recovering Your Oracle Real Application Clusters Database
  The Enterprise Manager Guided Recovery capability provides a Recovery Wizard that encapsulates the logic required for a wide range of restore and recovery scenarios.
- About Managing Your Database Backup Files
  You can use Oracle Enterprise Manager to manage your database backup files.
- Displaying Backup Reports for Your Oracle Real Application Clusters Database
  You can use Oracle Enterprise Manager to view backup reports.

> **See Also:**
>
> - *Oracle Database 2 Day DBA* for conceptual information about database backup and recovery
> - *Oracle Database Backup and Recovery User’s Guide* for more information about using the Recovery Manager utility

9.1 Overview of Oracle RAC Database Backup and Recovery

To protect your Oracle Real Application Clusters (Oracle RAC) database from hardware failures or disasters, you must have a physical copy of the database files.

The files protected by the backup and recovery facilities built into Oracle Enterprise Manager include data files, control files, server parameter files (SPFILEs), and archived redo log files. Using these files, your database can be reconstructed. The backup mechanisms that work at the physical level protect against damage at the file level, such as the accidental deletion of a data file or the failure of a disk drive. **Database recovery** involves restoring, or copying, the damaged files from backup and performing media recovery on the restored files. **Media recovery** is the application of redo logs or incremental backups to a restored data file to update it to the current time or some other specified time.

The Oracle Database flashback features, such as Oracle Flashback Drop and Oracle Flashback Table, provide a range of physical and logical data recovery tools as efficient, easy-to-use alternatives to physical and logical backup operations. The flashback features enable you to reverse the effects of unwanted database changes without restoring data files from backup or performing media recovery.

The Enterprise Manager physical backup and recovery features are built on the Recovery Manager (RMAN) command-line client. Enterprise Manager makes available many of the RMAN features, and provides wizards and automatic strategies to simplify and further automate RMAN-based backup and recovery.

> **Note:**
>
> For the RMAN utility to work properly on Linux platforms, the `$ORACLE_HOME/bin` directory must appear in the `PATH` variable before the `/usr/X11R6/bin` directory.

The Enterprise Manager Guided Recovery capability provides a Recovery Wizard that encapsulates the logic required for a wide range of file restoration and recovery scenarios, including the following:

- Complete restoration and recovery of the database
- Point-in-time recovery of the database or selected tablespaces
- Flashback Database
- Other flashback features of Oracle Database for logical-level repair of unwanted changes to database objects
- Media recovery at the block level for data files with corrupt blocks

If the database files are damaged or need recovery, then Enterprise Manager can determine which parts of the database must be restored from a backup and recovered, including early detection of situations such as corrupted database files. Enterprise Manager guides you through the recovery process, prompting for needed information and performing the required recovery actions.

See Also:

- "Performing Backups of Your Oracle Real Application Clusters Database"
- "Recovering Your Oracle Real Application Clusters Database"
- "About Managing Your Database Backup Files"
- *Oracle Database 2 Day DBA* for information about performing Oracle Database backup and recovery with Oracle Recovery Manager (RMAN)

9.2 About the Fast Recovery Area in Oracle RAC

Using a fast recovery area minimizes the need to manually manage disk space for your backup-related files and balance the use of space among the different types of files.

Oracle recommends that you enable a fast recovery area to simplify your backup management. The larger the fast recovery area is, the more useful it becomes. Ideally, the fast recovery area should be large enough to contain all the following files:

- A copy of all data files
- Incremental backups
- Online redo logs
- Archived redo log files that have not yet been backed up
- Control files and control file copies
- Autobackups of the control file and database initialization parameter file

The fast recovery area for an Oracle RAC database must be placed on an Oracle ASM disk group, a cluster file system, or on a shared directory that is configured through a network file system file for each Oracle RAC instance. In other words, the fast recovery area must be shared among all of the instances of an Oracle RAC database.

Note:

The preferred configuration for Oracle RAC is to use Oracle Automatic Storage Management (Oracle ASM) for storing the fast recovery area, using a different disk group for your recovery set than for your data files.

The location and disk quota must be the same on all instances. Oracle recommends that you place the fast recovery area on the shared Oracle ASM disks. In addition, you must set the `DB_RECOVERY_FILE_DEST` and `DB_RECOVERY_FILE_DEST_SIZE` parameters to the same values on all instances.

To use the fast recovery feature, you must first configure the fast recovery area for each instance in your Oracle RAC database.

See Also:

- "Creating Additional Oracle ASM Disk Groups"
- "Editing Initialization Parameter Settings for an Oracle RAC Database"
- *Oracle Database 2 Day DBA* for more information about the fast recovery area
- *Oracle Database 2 Day DBA* for information about how to configure backup and recovery for your database
- *Oracle Database Backup and Recovery User's Guide* for information about configuring the fast recovery area

9.3 Archiving the Oracle Real Application Clusters Database Redo Logs

To make your data highly available, it is important to configure the database so you can recover your data after a system failure. Redo logs contain a record of changes that were made to datafiles.

Redo logs are stored in redo log groups. You must have at least two redo log groups for your database.

After the redo log files in a group have filled up, the log writer process (LGWR) switches the writing of redo records to a new redo log group. Oracle Database can automatically save the inactive group of redo log files to one or more offline destinations, known collectively as the **archived redo log** (also called the archive log). The process of turning redo log files into archived redo log files is called **archiving**.

When you archive your redo log, you write redo log files to another location before they are overwritten. This location is called the archived redo log. These copies of redo log files extend the amount of redo data that can be saved and used for recovery. Archiving can be either enabled or disabled for the database, but Oracle recommends that you enable archiving.

- About Archived Redo Log Files for an Oracle RAC Database
  The primary consideration when configuring archiving is to ensure that all archived redo logs can be read from every node during recovery, and if possible during backups.
- About Parallelism and Backups Across Multiple RMAN Channels
  RMAN supports parallelism, which is the use of multiple channels and server sessions to perform the work of a single backup job or file restoration task.
- Configuring Archiving for Your Oracle RAC Database
  For Oracle RAC, each instance has its own thread of redo. The preferred configuration for Oracle RAC is to configure the fast recovery area using an Oracle ASM disk group that is separate from the Oracle ASM disk group used for your data files. Alternatively, you can use a cluster file system archiving scheme.

See Also:

- "Administering Redo Logs in Oracle RAC"
- *Oracle Database 2 Day DBA* for information about archived redo log files
- *Oracle Database 2 Day DBA* for information on planning space usage for the fast recovery area

9.3.1 About Archived Redo Log Files for an Oracle RAC Database

The primary consideration when configuring archiving is to ensure that all archived redo logs can be read from every node during recovery, and if possible during backups.

When you use Oracle Database Configuration Assistant (DBCA) to create your Oracle Real Application Clusters (Oracle RAC) database, each instance is configured with at least two redo log files that are stored in the shared storage area. If you have a two-node Oracle RAC database, then at least four redo logs are created for the database, two for each instance.

If you use a cluster file system to store the archived redo log files for your Oracle RAC database, then the redo log files are shared file system files. If you use Oracle ASM to store the archived redo log files for your Oracle RAC database, then each instance automatically has access to all the archived redo log files generated by the database. If you use shared storage to store the archived redo log files on each node, then you must configure the operating system to grant access to those directories for each instance of the cluster database that needs access to them.

During recovery, because the archived log destinations are visible from the node that performs the recovery, Oracle RAC can successfully recover the archived redo log data. For creating backups of your Oracle RAC database, the strategy that you choose depends on how you configure the archiving destinations for each node. Whether only one node or all nodes perform archived redo log backups, you must ensure that the archived redo logs for every instance are backed up.

To backup the archived redo logs from a single node, that node must have access to the archived log files of the other instances. The archived redo log naming scheme that you use is important because when a node writes to a log with a specific file name on its file system, the file must be readable by any node that must access this archived redo log. For example, if `node1` archives a log to `/oracle/arc_dest/log_1_100_23452345.arc`, then `node2` can back up this archived redo log only if it can read`/oracle/arc_dest/log_1_100_23452345.arc` on its own file system.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for information about creating archived log files in Oracle ASM
- *Oracle Real Application Clusters Administration and Deployment Guide* for information about naming conventions for archived redo log files in Oracle RAC

9.3.2 About Parallelism and Backups Across Multiple RMAN Channels

RMAN supports parallelism, which is the use of multiple channels and server sessions to perform the work of a single backup job or file restoration task.

Recovery Manager (RMAN) depends on server sessions, processes that run on the database server, to perform backup and recovery tasks. Each server session in turn corresponds to an RMAN channel, representing one stream of data to or from a backup device.

Because the Oracle Database control file, SPFILE, and data files are accessible by any instance, the backup operation of these files is distributed across all the allocated channels. For backups of the archived redo log, the actions performed by RMAN depend on the type of archiving scheme used by your Oracle RAC database.

If you use a local archiving scheme, then each instance writes the archived redo log files to a local directory. When multiple channels are allocated that have access to the archived redo log, for each archived redo log file, RMAN determines which channels have access to that archived redo log file. Then, RMAN groups the archived redo log files that can be accessed by a channel and schedules a backup job using that channel.

If each node in the cluster writes the archived redo log files to Oracle ASM, a clustered file system, or other type of shared storage, then each instance has access to all the archived redo log files. In this case, the backup of the archived redo log is distributed across all the allocated channels.

See Also:

- "About Archived Redo Log Files for an Oracle RAC Database"
- "Performing Backups of Your Oracle Real Application Clusters Database"
- *Oracle Database 2 Day DBA* for more information about RMAN backups
- *Oracle Database 2 Day DBA* for more information about configuring backup device settings

9.3.3 Configuring Archiving for Your Oracle RAC Database

For Oracle RAC, each instance has its own thread of redo. The preferred configuration for Oracle RAC is to configure the fast recovery area using an Oracle ASM disk group that is separate from the Oracle ASM disk group used for your data files. Alternatively, you can use a cluster file system archiving scheme.

To configure archiving for your Oracle RAC database:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page, and log in as a SYSDBA user.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Availability** menu, select **Backup and Recovery**, and then **Recovery Settings**.

   The Recovery Settings page appears.

3. In the Media Recovery section, select the **ARCHIVELOG mode** option.

4. In the **Log Archive Filename Format** field, accept the default value, or enter the desired format.

   For clustered databases, the format for the archive log file name should contain the `%t` modifier, to indicate which redo log thread the archived redo log file belongs to. As a best practice, the format for the archive log file name should also include the `%s` (log sequence number) and `%r` (resetlogs identifier) modifiers.

5. If the archive log destination is the same for all instances, then in the **Archive Log Destination** field, change the value to the location of the archive log destination for the cluster database.

   For example, you might set it to `+DATA` if using Oracle ASM, or to `/u01/oradata/arch` if you want local archiving on each node.

   If you must configure a different archive log destination for any instance, then you must go to the Initialization Parameters page and modify the `LOG_ARCHIVE_DEST_1` parameter that corresponds to the instance for which you want to configure the archive log destination. The Instance column should display the name of the instance, for example `sales1`. Change the **Value** field to contain the location of the archive log destination for that instance.

6. If you want to configure multiple archive log destinations for the database, then on the Recovery Settings page, click **Add Another Row** under the Archive Log Destination field.

7. After you have finished configuring archiving, click **Apply**.

8. When prompted to restart the database, click **Yes**.

9. Enter the host and SYSDBA user credentials, then click **Continue**.

10. Wait a couple of minutes, then click **Refresh**.

    If the database has been restarted, then you are prompted to enter the login credentials.

> **See Also:**
>
> - "About Archived Redo Log Files for an Oracle RAC Database"
> - "About Configuring Initialization Parameters for an Oracle RAC Database
>   "
> - "Editing Initialization Parameter Settings for an Oracle RAC Database"
> - *Oracle Database 2 Day DBA* for information about configuring settings used for instance recovery, media recovery, and fast recovery
> - *Oracle Real Application Clusters Administration and Deployment Guide* for more information about configuring and managing archived redo log files for an Oracle RAC database

9.4 About Preparing for Backup and Recovery Operations

Before taking backups of your Oracle Real Application Clusters (Oracle RAC) database using Enterprise Manager, you must configure access for the user performing the backups, or **credentials**.

You can also configure the default values for certain backup settings, so they do not have to be specified every time a backup is taken.

- About Credentials for Performing Backup and Recovery
  When using Enterprise Manager, you must have the proper credentials to perform the configuration tasks for backup and recovery, to schedule backup jobs, and to perform recovery.
- About Configuring Backup Settings
  You can configure several settings and policies for backups.

> **See Also:**
>
> "About Operating System Users and Groups"

9.4.1 About Credentials for Performing Backup and Recovery

When using Enterprise Manager, you must have the proper credentials to perform the configuration tasks for backup and recovery, to schedule backup jobs, and to perform recovery.

The following credentials may be required:

- The Oracle database administrator user you use when you log in to Enterprise Manager
- The host operating system user whose credentials you provide when performing backup and recovery tasks

To perform or schedule RMAN tasks, you must either log in to Enterprise Manager as a user with `SYSDBA` or `SYSBACKUP` administrative privilege, or provide host operating system credentials for a user who is a member of the `dba` group. The host operating system user must also have execute permission for the RMAN command-line client.

For tasks requiring host operating system credentials, a Host Credentials form appears at the bottom of the page used to perform the task. Enterprise Manager uses the credentials when it invokes RMAN to perform jobs you requested or scheduled.

The Host Credentials form always includes an option labeled Save as Preferred Credential. If you select this option before performing your action, then the provided credentials are stored persistently for the currently logged-in Oracle database user. The preferred credentials are reused by default whenever you log in as that user and perform operations requiring host credentials.

See Also:

- "About Operating System Users and Groups"
- "About Configuring User Accounts"
- *Oracle Database 2 Day DBA* for information about the user credentials required to perform backup and recovery

9.4.2 About Configuring Backup Settings

You can configure several settings and policies for backups.

Assuming you have a fast recovery area configured, you can configure several settings and policies that determine how backups are stored, which data is backed up, and how long backups are retained before being purged from the fast recovery area. You can also configure settings to optimize backup performance for your environment.

See Also:

- "Configuring Archiving for Your Oracle RAC Database"
- "Overview of Oracle RAC Database Backup and Recovery"
- *Oracle Database 2 Day DBA* for more information about configuring backup policy settings
- *Oracle Database 2 Day DBA* for more information about configuring backup settings

9.5 Performing Backups of Your Oracle Real Application Clusters Database

You can use Oracle Enterprise Manager to backup your Oracle Real Application Clusters (Oracle RAC) database.

When you use Oracle ASM to manage database files, Oracle recommends that you use RMAN for creating backups.

You must have both database (`SYSDBA`) privileges and host operating system (`OSDBA`) credentials to perform backup and recovery operations. If you log in to Enterprise Manager with `SYSDBA` privileges, then any operating system user who has execute permission for the RMAN command-line client can perform backups of an Oracle Real Application Clusters (Oracle RAC) database. However, if you log in as a database user without `SYSDBA` privileges, then you must provide the name and password of an operating system user that is a member of the `OSDBA` group before you can perform the backup operation.

To back up an Oracle RAC database:

1. On the Cluster Database Home page, select **Availability**.

   The Cluster Database Availability page appears.

2. In the Backup/Recovery section, under the heading Manage, select **Schedule Backup**.
3. Follow the backup procedures outlined in Chapter 9, "Performing Backup and Recovery" of *Oracle Database 2 Day DBA* or click **Help** on this page.

- Backing Up Archived Redo Log Files
  You can configure RMAN to automatically delete the archived redo log files from disk after they have been safely backed up.

> **See Also:**
>
> - "About Operating System Users and Groups"
> - "About Credentials for Performing Backup and Recovery "

9.5.1 Backing Up Archived Redo Log Files

You can configure RMAN to automatically delete the archived redo log files from disk after they have been safely backed up.

This feature helps to reduce the disk space used by your Oracle RAC database, and prevents unnecessary outages that might occur if you run out of available disk space.

Whether only one node or all nodes perform archive log backups, ensure that all archived redo log files for all nodes are backed up. If you use a local archiving scheme, then allocate multiple channels to provide RMAN access to all the archived redo log files.

To configure RMAN to automatically delete the archived redo log file files from disk after they have been safely backed up, when creating or scheduling your database backups:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page, and log in as a SYSDBA user.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.
2. From the **Availability** menu, select **Backup and Recovery**, and then **Schedule Backup**.
3. Choose a backup type and click **Schedule Customized Backup**.
4. While specifying the options for your backup, select **Also back up all archived logs** on disk if you are performing an online backup. There is no need to back up archived redo log files when performing an offline backup because the database is in a consistent state at the time of backup and does not require media recovery if you restore.
5. Select **Delete all archived logs from disk after they are successfully backed up** if you are using shared storage for your archived redo log files.

> **Note:**
>
> Do not select **Delete all archived logs from disk after they are successfully backed up** if you are using a fast recovery area as your *only* archive log destination. In this case, archived redo log files that have been backed up are deleted automatically as space is needed for storage of other files.

> **See Also:**
>
> - "About Archived Redo Log Files for an Oracle RAC Database"
> - *Oracle Database 2 Day DBA* for more information about configuring backup settings

9.6 Recovering Your Oracle Real Application Clusters Database

The Enterprise Manager Guided Recovery capability provides a Recovery Wizard that encapsulates the logic required for a wide range of restore and recovery scenarios.

Enterprise Manager can determine which parts of the database must be restored and recovered, including early detection of situations such as corrupted database files. Enterprise Manager takes you through the recovery process, prompting for information and performing required file restoration and recovery actions.

This section discusses both instance recovery and media recovery.

- About Preparing to Restore and Recover Your Oracle RAC Database
  The node that performs the recovery of an Oracle Real Application Clusters (Oracle RAC) database must be able to restore all the required data files. That node must also be able to either read all the required archived redo log files on disk or be able to restore the archived redo log files from backup files.
- About Restoring Archived Redo Log File Files
  If you use Oracle ASM or a clustered file system for storing the archived redo log files, then any instance can restore the archived redo log files.
- About Performing Recovery Using Parallelism
  Oracle Real Application Clusters (Oracle RAC) automatically selects the optimum degree of parallelism for instance failure and media recovery.
- Recovering the Oracle RAC Database Using Enterprise Manager
  You can use Oracle Enterprise Manager and Recovery Manager (RMAN) to restore and recover an Oracle Real Application Clusters (Oracle RAC) database.
- Recovering the Server Parameter File from an Automatic File Backup
  You can use Enterprise Manager to recover a lost or damaged server parameter file (SPFILE).

> **See Also:**
>
> - "Overview of Oracle RAC Database Backup and Recovery"
> - "About the Fast Recovery Area in Oracle RAC"

9.6.1 About Preparing to Restore and Recover Your Oracle RAC Database

The node that performs the recovery of an Oracle Real Application Clusters (Oracle RAC) database must be able to restore all the required data files. That node must also be able to either read all the required archived redo log files on disk or be able to restore the archived redo log files from backup files.

- About Configuring Access to the Archive Log
  During recovery, because the archive log file destinations are visible from the node that performs the recovery, Oracle RAC can successfully access the archived redo log files during recovery.
- About Putting the Oracle RAC Database Instances into the Correct State
  If your database requires media recovery, you must start the database instances in the correct mode before beginning recovery.

See Also:

- "About Archived Redo Log Files for an Oracle RAC Database"
- *Oracle Database 2 Day DBA* for information about database backup and recovery concepts
- *Oracle Database 2 Day DBA* for information about archived redo log files

9.6.1.1 About Configuring Access to the Archive Log

During recovery, because the archive log file destinations are visible from the node that performs the recovery, Oracle RAC can successfully access the archived redo log files during recovery.

If you do not use shared storage or a clustered file system to store the archived redo log files for your cluster database, then you must make the archived redo log files available to the node performing the recovery.

See Also:

- "About Archived Redo Log Files for an Oracle RAC Database"
- "Administering Redo Logs in Oracle RAC"
- *Oracle Database 2 Day DBA* for information about the online redo log files
- *Oracle Database 2 Day DBA* for information about archived redo log files

9.6.1.2 About Putting the Oracle RAC Database Instances into the Correct State

If your database requires media recovery, you must start the database instances in the correct mode before beginning recovery.

Recovery of a failed instance in Oracle RAC is automatic. If an Oracle RAC database instance fails, then a surviving database instance processes the online redo logs generated by the failed instance to ensure that the database contents are in a consistent state. When recovery completes, Oracle Clusterware attempts to restart the failed instance automatically.

Media recovery is a manual process that occurs while a database is closed. A media failure is the failure of a read or write operation of a disk file required to run the database, due to a physical problem with the disk such as a head malfunction. Any database file can be vulnerable to a media failure. If a media failure occurs, then you must perform media recovery to restore and recover the damaged database files. Media recovery is always done by one instance in the cluster.

Before starting media recovery, the instance that is performing the recovery should be started in `MOUNT` mode. The other instances should be started in `NOMOUNT` mode.

See Also:

- "Starting and Stopping Oracle RAC Databases and Database Instances"
- *Oracle Database 2 Day DBA* for information about the Oracle advised recovery feature and Data Recovery Advisor

9.6.2 About Restoring Archived Redo Log File Files

If you use Oracle ASM or a clustered file system for storing the archived redo log files, then any instance can restore the archived redo log files.

During a restore operation, RMAN automatically locates the most recent backups of the database that are available. A channel connected to a specific node attempts to restore files that were backed up only to that node. For example, assume that an archived redo log file with the sequence number 1_001 is backed up to a device attached to the node `racnode1`, while the archived redo log file with sequence number 2_003 is backed up to a device attached to the node `racnode2`. If you allocate channels that connect to nodes `racnode1` and `racnode2` for a restore operation, then the channel connected to `racnode1` restores log sequence 1_001, but not log sequence 2_003. The channel connected to `racnode2` can restore log sequence 2_003, but not log sequence 1_001.

See Also:

- "About Archived Redo Log Files for an Oracle RAC Database"
- "Configuring Archiving for Your Oracle RAC Database"
- *Oracle Database Backup and Recovery User's Guide* for more information about restoring archived redo log file files

9.6.3 About Performing Recovery Using Parallelism

Oracle Real Application Clusters (Oracle RAC) automatically selects the optimum degree of parallelism for instance failure and media recovery.

When using Enterprise Manager and Recovery Manager (RMAN) to perform the recovery, Oracle RAC automatically makes parallel the following three stages of recovery:

- Restoring Data files—When restoring data files, the number of channels you allocate in the RMAN recovery script effectively sets the parallelism that RMAN uses. For example, if you allocate five channels, then you can have up to five parallel streams restoring data files.
- Applying Incremental Backups—Similarly, when you are applying incremental backups, the number of channels you allocate determines the potential parallelism.

- Applying Archived Redo Log Files—Using RMAN, the application of archived redo log files is performed in parallel. Oracle RAC automatically selects the optimum degree of parallelism based on available CPU resources.

See Also:

- "Overview of Oracle RAC Database Backup and Recovery"
- *Oracle Database 2 Day DBA* for more information about incremental backups of data files
- *Oracle Database 2 Day DBA* for more information about configuring recovery settings

9.6.4 Recovering the Oracle RAC Database Using Enterprise Manager

You can use Oracle Enterprise Manager and Recovery Manager (RMAN) to restore and recover an Oracle Real Application Clusters (Oracle RAC) database.

When using Enterprise Manager and RMAN, the process of recovering and restoring an Oracle RAC database is essentially the same as for a single-instance Oracle database, except that you access RMAN from the Availability page at the cluster database level, instead of at the instance level.

To use Enterprise Manager and RMAN to restore and recover an Oracle RAC database:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page, and log in as a SYSDBA user.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Availability** menu, select **Backup and Recovery**, and then **Perform Recovery**.

   The Perform Recovery page appears.

3. Follow the recovery procedures outlined in Chapter 9 of *Oracle Database 2 Day DBA*.

See Also:

- "About Preparing to Restore and Recover Your Oracle RAC Database"
- *Oracle Database 2 Day DBA* for more information about performing user-directed recovery

9.6.5 Recovering the Server Parameter File from an Automatic File Backup

You can use Enterprise Manager to recover a lost or damaged server parameter file (SPFILE).

To recover an SPFILE for an Oracle RAC database:

1. Start the database in `MOUNT` mode.
2. In Oracle Enterprise Manager, go to the Cluster Database Home page, and log in as a SYSDBA user.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.
3. From the **Availability** menu, select **Backup and Recovery**, and then **Perform Recovery**.

   When the database is not open, the Perform Recovery link takes you to the SPFILE restore page.
4. Specify the location of the fast recovery area, if configured.
5. In the Backup Information section, select **Use Other Backup Information** and **Use an Autobackup**.
6. On the Perform Recovery: Restore SPFILE page, specify a different location for the SPFILE to be restored to.
7. When finished selecting your options, click **Restore**, then click **Yes** to confirm you want to restore the SPFILE.
8. After the SPFILE is restored, you are prompted to log in to the database again.

> **See Also:**
>
> - "Starting and Stopping Oracle RAC Databases and Database Instances"
> - "About the Server Parameter File for Oracle Real Application Clusters"
> - *Oracle Database Backup and Recovery User’s Guide* for more information about recovering a server parameter file

9.7 About Managing Your Database Backup Files

You can use Oracle Enterprise Manager to manage your database backup files.

Managing RMAN backup files, with or without Enterprise Manager, consists of two tasks:

1. Managing the backup files for your database that are stored on disk or tape
2. Managing the record of those backup files in the RMAN repository

Enterprise Manager simplifies both backup file management tasks. Other tasks involved in managing backup files include the following:

- Searching for backup files
- Validating the contents of backup sets or image copies
- Cross-checking a backup
- Deleting expired or obsolete backup files
- Marking backup files as available or unavailable

> **See Also:**
>
> - "Overview of Oracle RAC Database Backup and Recovery"
> - "Performing Backups of Your Oracle Real Application Clusters Database"
> - *Oracle Database 2 Day DBA* for more information about managing RMAN backup files
> - *Oracle Database 2 Day DBA* for information about backup reports
> - *Oracle Database 2 Day DBA* for information about validating your backups and testing your backup strategy
> - *Oracle Database 2 Day DBA* for information about deleting expired backups
> - *Oracle Database 2 Day DBA* for information about marking backups as available or unavailable

9.8 Displaying Backup Reports for Your Oracle Real Application Clusters Database

You can use Oracle Enterprise Manager to view backup reports.

Backup reports contain summary and detailed information about past backup jobs run by Recovery Manager (RMAN), including backup jobs run through Oracle Enterprise Manager and the RMAN command-line client.

To view backup reports:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page, and log in as a SYSDBA user.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Availability** menu, select **Backup and Recovery**, and then **Backup Reports**.

   The View Backup Report page appears, with a list of recent backup jobs.

3. In the Search section, specify any filter conditions and click **Go** to restrict the list to backups of interest.

You can use the **Search** section of the page to restrict the backups listed by the time of the backup, the type of data backed up, and the status of the jobs (whether it succeeded or failed, and whether warnings were generated during the job).

4. To view detailed information about any backup, click the backup job name in the **Backup Name** column.

   The Backup Report page is displayed for the selected backup job. This page contains summary information about this backup job, such as how many files of each type were backed up, the total size of the data backed up, and the number, size, and type of backup files created.

   The Backup Report page also contains a Search section that you can use to quickly run a search for another backup job or backup jobs from a specific date range. The resulting report contains aggregate information for backup jobs matching the search criteria.

> **See Also:**
>
> - "About Managing Your Database Backup Files"
> - "Performing Backups of Your Oracle Real Application Clusters Database"
> - *Oracle Database 2 Day DBA* for more information about displaying backup reports

10
Monitoring and Troubleshooting Your Cluster

You can use the Autonomous Health Framework to monitor and troubleshoot your cluster.

- Autonomous Health Framework
  Autonomous Health Framework is a collection of tools that can analyze the collected data, and proactively identify issues before they affect the health of your clusters or your Oracle RAC databases.
- Monitoring Oracle Clusterware Using Enterprise Manager
  You can use Enterprise Manager to monitor Oracle Clusterware.
- Troubleshooting Configuration Problems in Oracle RAC Environments
  The Cluster Ready Services Control (CRSCTL) command `check` enables you to determine the status of several Oracle Clusterware components simultaneously.

> **See Also:**
>
> - *Oracle Real Application Clusters Administration and Deployment Guide* for more information about diagnosing problems for Oracle Real Application Clusters components
> - *Oracle Clusterware Administration and Deployment Guide* for more information about diagnosing problems for Oracle Clusterware components

10.1 Autonomous Health Framework

Autonomous Health Framework is a collection of tools that can analyze the collected data, and proactively identify issues before they affect the health of your clusters or your Oracle RAC databases.

The majority of the tools that make up Autonomous Health Framework are already available in Oracle Database 12*c* release 1. In Oracle Database 12*c* release 2, the output of several tools is consolidated in the Grid Infrastructure Management Repository and analyzed in real-time to detect patterns which can indicate the development of issues on the production clusters.

- Cluster Verification Utility (CVU)
  The Cluster Verification Utility (CVU) can assist you in diagnosing a wide variety of configuration problems. As part of Autonomous Health Framework, CVU can perform these checks periodically by running autonomously on a scheduled basis.

- ORAchk
  As part of Autonomous Health Framework, ORAchk is configured to run in Daemon Mode. ORAchk proactively scans for known problems within a wide range of products and autonomously reports on these issues.
- Cluster Health Monitor
  Cluster Health Monitor is integrated part of Oracle Grid Infrastructure. Cluster Health Monitor collects operating system statistics using operating system APIs to gain performance and reduce the CPU usage overhead.
- Cluster Health Advisor
  The Cluster Health Advisor daemon collects data from Oracle Real Application Clusters (Oracle RAC), Oracle RAC One Node databases, and the host operating system and hardware resources, to analyze and produce a root cause and action after finding a database or host performance issue.
- Trace File Analyzer Collector
  Trace File Analyzer centralizes and automates the collection of diagnostic information.
- Hang Manager
  Hang Manager can reliably detect and resolve hung database sessions in a timely manner.
- Managing Memory Pressure for Database Servers
  Memory Guard detects memory pressure on a server in real time and redirects new sessions to other servers to prevent using all available memory on the stressed server.
- Oracle Database Quality of Service Management
  Oracle Database Quality of Service (QoS) Management is an automated, policy-based product that monitors the workload requests for an entire system.

10.1.1 Cluster Verification Utility (CVU)

The Cluster Verification Utility (CVU) can assist you in diagnosing a wide variety of configuration problems. As part of Autonomous Health Framework, CVU can perform these checks periodically by running autonomously on a scheduled basis.

Cluster Verification Utility (CVU) performs system checks in preparation for installation, patch updates, or other system changes. Using CVU ensures that you have completed the required system configuration and preinstallation steps so that your Oracle Grid Infrastructure or Oracle Real Application Clusters (Oracle RAC) installation, update, or patch operation, completes successfully.

Oracle Universal Installer is fully integrated with CVU, automating many CVU prerequisite checks. Oracle Universal Installer runs all prerequisite checks and associated fixup scripts when you run the installer.

- Verifying the Existence of Node Applications
  You use the CVU `comp nodeapp` command to verify the existence of node applications, namely the virtual IP (VIP), Oracle Notification Services (ONS), and Global Service Daemon (GSD), on all the nodes.
- Verifying the Integrity of Oracle Clusterware Components
  You use the CVU `comp crs` command to verify the existence of all the Oracle Clusterware components.

- Verifying the Integrity of the Oracle Cluster Registry
  You use the CVU `comp ocr` command to verify the integrity of the Oracle Clusterware registry.
- Verifying the Integrity of Your Entire Cluster
  Use the CVU `comp clu` command to check that all nodes in the cluster have the same view of the cluster configuration.
- Checking the Settings for the Interconnect
  Use the CVU `comp nodereach` or `comp nodecon`command to check the settings for the interconnect.
- Enabling Tracing
  CVU does not generate trace files unless you enable tracing.

> **See Also:**
>
> *Oracle Clusterware Administration and Deployment Guide*

10.1.1.1 Verifying the Existence of Node Applications

You use the CVU `comp nodeapp` command to verify the existence of node applications, namely the virtual IP (VIP), Oracle Notification Services (ONS), and Global Service Daemon (GSD), on all the nodes.

To verify the existence of node applications:

1. In a command window, log in to the operating system as the user who owns the Oracle Clusterware software installation.
2. Use the `comp nodeapp` command of CVU, using the following syntax:

   ```
   cluvfy comp nodeapp [ -n node_list] [-verbose]
   ```

   where *`node_list`* represents the nodes to check.
3. If the `cluvfy` command returns the value of `UNKNOWN` for a particular node, then CVU cannot determine whether a check passed or failed.

   Determine if the failure was caused by one of the following reasons:

   - The node is down.
   - Executable files that CVU requires are missing in the *`CRS_home`*`/bin` directory or the *`Oracle_home`*`/bin` directory.
   - The user account that ran CVU does not have permissions to run common operating system executable files on the node.
   - The node is missing an operating system patch or required package.
   - The kernel parameters on that node were not configured correctly and CVU cannot obtain the operating system resources required to perform its checks.

See Also:

- "Troubleshooting Configuration Problems in Oracle RAC Environments"
- *Oracle Clusterware Administration and Deployment Guide* for more information about the Cluster Verification Utility

10.1.1.2 Verifying the Integrity of Oracle Clusterware Components

You use the CVU `comp crs` command to verify the existence of all the Oracle Clusterware components.

To verify the integrity of Oracle Clusterware components:

1. In a command window, log in to the operating system as the user who owns the Oracle Clusterware software installation.
2. Use the `comp crs` command of CVU, using the following syntax:

   ```
   cluvfy comp crs [ -n node_list] [-verbose]
   ```

   where *`node_list`* represents the nodes to check.

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for more information about the Cluster Verification Utility

10.1.1.3 Verifying the Integrity of the Oracle Cluster Registry

You use the CVU `comp ocr` command to verify the integrity of the Oracle Clusterware registry.

To verify the integrity of the Oracle Clusterware registry:

1. In a command window, log in to the operating system as the user who owns the Oracle Clusterware software installation.
2. Use the `comp ocr` command of CVU, using the following syntax:

   ```
   cluvfy comp ocr [ -n node_list] [-verbose]
   ```

   where *node_list* represents the nodes to check.

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for more information about the Cluster Verification Utility

10.1.1.4 Verifying the Integrity of Your Entire Cluster

Use the CVU `comp clu` command to check that all nodes in the cluster have the same view of the cluster configuration.

To verify the integrity of your cluster:

1. In a command window, log in to the operating system as the user who owns the Oracle Clusterware software installation.
2. Use the `comp clu` command of CVU, using the following syntax:

   ```
   cluvfy comp clu [-verbose]
   ```

> **See Also:**
>
> - *Oracle Clusterware Administration and Deployment Guide* for more information about the Cluster Verification Utility
> - "Verifying Your Oracle RAC Database Installation"

10.1.1.5 Checking the Settings for the Interconnect

Use the CVU `comp nodereach` or `comp nodecon`command to check the settings for the interconnect.

Cache Fusion enhances the performance of Oracle RAC by utilizing a high-speed interconnect to send data blocks to another instance's buffer cache. The high-speed interconnect should be a private network with the highest bandwidth to maximize performance.

For network connectivity verification, CVU discovers all the available network interfaces if you do not specify an interface on the CVU command line.

To check the settings for the interconnect:

1. In a command window, log in to the operating system as the user who owns the Oracle Clusterware software installation.
2. To verify the accessibility of the cluster nodes, specified by *node_list*, from the local node or from any other cluster node, specified by `srcnode`, use the component verification command `nodereach` as follows:

   ```
   cluvfy comp nodereach -n node_list [ -srcnode node ] [-verbose]
   ```

3. To verify the connectivity among the nodes, specified by *node_list*, through the available network interfaces from the local node or from any other cluster node, use the `comp nodecon` command as shown in the following example:

   ```
   cluvfy comp nodecon -n node_list -verbose
   ```

   When you issue the `nodecon` command as shown in the previous example, it instructs CVU to perform the following tasks:

- Discover all the network interfaces that are available on the specified cluster nodes.
- Review the corresponding IP addresses and subnets for the interfaces.
- Obtain the list of interfaces that are suitable for use as VIPs and the list of interfaces to private interconnects.
- Verify the connectivity among all the nodes through those interfaces.

When you run the `nodecon` command in verbose mode, it identifies the mappings between the interfaces, IP addresses, and subnets.

4. To verify the connectivity among the nodes through specific network interfaces, use the `comp nodecon` command with the `-i` option and specify the interfaces to be checked with the *interface_list* argument:

```
cluvfy comp nodecon -n node_list -i interface_list [-verbose]
```

For example, you can verify the connectivity among the nodes `racnode1`, `racnode2`, and `racnode3`, through the specific network interface `eth0` by running the following command:

```
cluvfy comp nodecon -n racnode1,racnode2,racnode3 -i eth0 -verbose
```

See Also:

- "Configuring the Network"
- *Oracle Clusterware Administration and Deployment Guide* for more information about the Cluster Verification Utility

10.1.1.6 Enabling Tracing

CVU does not generate trace files unless you enable tracing.

The CVU trace files are created in the `ORACLE_BASE/crsdata/`*`host_name`*`/cvu` directory by default. Oracle Database automatically rotates the log files and the most recently created log file has the name `cvutrace.log.0`. You should remove unwanted log files or archive them to reclaim disk place if needed.

To enable tracing using CVU:

1. In a command window, log in to the operating system as the `root` user.
2. Set the environment variable `SRVM_TRACE` to `true`.

```
# set SRVM_TRACE=true; export SRVM_TRACE
```

3. Run the command to trace.

See Also:

Oracle Clusterware Administration and Deployment Guide for more information about enabling tracing

10.1.2 ORAchk

As part of Autonomous Health Framework, ORAchk is configured to run in Daemon Mode. ORAchk proactively scans for known problems within a wide range of products and autonomously reports on these issues.

ORAchk replaces the popular RACcheck tool, extending the coverage based on prioritization of top issues reported by users. You can configure ORAchk to upload its findings to the Collection Manager Repository for centralized examination and analysis.

To obtain the most out of ORAchk, Oracle recommends that you run ORAchk as the `root` user (via password or SUDO).

- Overview of Oracle ORAchk
  Oracle ORAchk provides a lightweight and non-intrusive health check framework for the Oracle stack of software and hardware components.
- Running ORAchk
  You can schedule ORAchk to run at periodic intervals or you can run it as needed.
- ORAchk HTML Report Output
  Oracle ORAchk generates a detailed HTML report with findings and recommendations.

10.1.2.1 Overview of Oracle ORAchk

Oracle ORAchk provides a lightweight and non-intrusive health check framework for the Oracle stack of software and hardware components.

Oracle ORAchk is provided as a value add-on to your existing support contract. There is no additional fee or license required to run Oracle ORAchk.

Features of ORAchk

- Automates risk identification and proactive notification before business is impacted.
- Runs health checks based on critical, reoccurring problems across Oracle customer base.
- Runs in your environment with no need to send anything to Oracle.
- Enables you to schedule email health check reports.
- Integrates the findings into other tools of your choice.

The scope of Oracle ORAchk increases with new releases. The following is the current scope:

Oracle Database

- Single-instance Oracle Database

- Oracle Grid Infrastructure and Oracle RAC
- Maximum Availability Architecture (MAA) validation
- Upgrade Readiness validation
- Oracle GoldenGate
- Application Continuity

Enterprise Manager Cloud Control (12c only)

- Management Repository
- Management Agents
- Oracle Management Service (OMS), version 12.1.0.1 and later on Linux only

Oracle Hardware Systems

- Oracle Solaris
- Oracle Solaris Cluster
- Oracle Systems configuration for Oracle Middleware and Oracle Applications
- Oracle ZFS Storage Appliance
- Oracle Virtual Networking

10.1.2.2 Running ORAchk

You can schedule ORAchk to run at periodic intervals or you can run it as needed.

Oracle recommends to use the daemon process to schedule recurring health checks at regular intervals.

Configure the daemon to:

- Schedule recurring health checks at regular interval.
- Send email notifications when the health check runs complete, clearly showing any differences since the last run.
- Purge collection results after a pre-determined period.
- Check and send email notification about stale passwords.
- Store multiple profiles for automated health check runs.
- Restart automatically if the server or node where it is running restarts.

Oracle recommends that you run on-demand health checks in the following scenarios:

- Pre or post-upgrades
- Machine moves from one subnet to another
- Hardware failure or repair
- Problem troubleshooting
- In addition to go-live testing

1. To run on-demand health checks, do not use the daemon process started by any other user. Run on-demand health checks within the same directory where you have started the daemon.

   ```
   $ ./orachk
   ```

2. To schedule health checks, do the following:
   a. Set the daemon properties using the `-set` option.

      At a minimum, set `AUTORUN_SCHEDULE` and `NOTIFICATION_EMAIL`.

      For example, to set the tool to run at 3 AM every Sunday and email the results to some.body@acompany.com, run the following command:

      ```
      $ ./orachk –set "AUTORUN_SCHEDULE=3 * *
      0 ;NOTIFICATION_EMAIL=some.body@example.com"
      ```
   b. Configure the health check daemon as described in Automated Daemon Mode Operation.
   c. Start the daemon as `root` (recommended) or as the Oracle Database or the Oracle Grid Infrastructure home owner.

      ```
      $ ./orachk –d start
      ```
   d. Answer the questions prompted during startup.

Oracle ORAchk generates a detailed HTML report with findings and recommendations.

For the subsequent health check runs after the first email notification, the daemon emails the diff report of the current run and the immediate past run to all users specified in the NOTIFICATION_EMAIL list.

The email notification message contains:

- System Health Score of this run compared to previous.
- Summary of number of checks run and differences between runs.
- Most recent report result as attachment.
- Previous report result as attachment.
- Diff report as attachment.

> **See Also:**
>
> *Oracle Autonomous Health Framework User's Guide* for more information about using and configuring ORAchk.

10.1.2.3 ORAchk HTML Report Output

Oracle ORAchk generates a detailed HTML report with findings and recommendations.

The Health Check HTML report contains the following:

- High level health score.
- Summary of the run.
- Table of contents that provides easy access to findings.
- Findings and recommendations to resolve the issues.

System Health Score and Summary

Oracle ORAchk calculates a high level System Health Score based on the number of checks that passed or failed. A summary of the run shows, which includes the following information:

- The cluster name
- The operating system and software versions
- The EM Agent home directory
- Cluster information — the number of nodes, the number of database servers, the number of storage servers, and the number of IB switches
- The version of ORAchk
- The name of the collection file
- The duration of the ORAchk run
- The user that started ORAchk
- The date that ORAchk was run

HTML Report Table of Contents and Features

The **Table of Contents** provides links to each of the major sections within the HTML report.

The **Report Features** allow you to:

- Filter checks based on their statuses.
- Select the regions.
- Expand or collapse all checks.
- View check IDs.
- Remove findings from the report.
- Get a printable view.

HTML Report Findings

Report findings are grouped by Oracle Stack component. Findings include:

- Status of check (`FAIL`, `WARNING`, `INFO`, or `PASS`).
- Type of check.
- Check message.
- Location where the check was run.
- Link to expand details for further findings and recommendations.

For any of the findings you can click **View** to see the findings and the recommendations, which can include:

- Solution to solve the problem.
- Applicable recommendations.
- Where the problem does not apply.

- Links to relevant documentation or My Oracle Support Notes.
- Example of data the recommendation is based on.

Maximum Availability Architecture (MAA) Scorecard

The Maximum Availability Architecture (MAA) Scorecard is displayed after the Findings group.

The MAA Scorecard provides a set of best practices for maximum availability architecture. It also shows results related to maximum availability, such as the installed software versions checked for noncurrent software, and use of incompatible features.

Findings Needing Further Review

Issues that health checks have only a partial view and need user reviews to determine if they are relevant are displayed in the Findings needing further review section.

Platinum Certification

The Platinum Certification section shows a list of compliance status items for the Oracle Platinum service. For the existing Platinum customers it is a review. For customers not yet participating in Oracle Platinum, it is an indication of readiness to participate in Oracle Platinum.

Clusterwide Linux Operating system Health Check (VMPScan)

A summary of the VMPScan report is shown in the Clusterwide Linux Operating System Health Check (VMPScan) section of the report.

The full VMPScan report is also available within the collection/reports and collection/outfiles/vmpscan directory.

File Attribute Changes

The File Attribute Changes section is shown in the report only when you run Oracle ORAchk with the `–fileattr` option.

Skipped Checks

Any checks that were not able to be run and skipped for some reason are shown in the Skipped Checks section.

Component Elapsed Times

The Component Elapsed Times gives a breakdown of time required to check various components.

This can be useful when diagnosing performance problems.

Top 10 Time Consuming Checks

The Top 10 Time Consuming Checks section shows the slowest 10 checks that were run.

This can be useful when diagnosing performance problems.

10.1.3 Cluster Health Monitor

Cluster Health Monitor is integrated part of Oracle Grid Infrastructure. Cluster Health Monitor collects operating system statistics using operating system APIs to gain performance and reduce the CPU usage overhead.

Cluster Health Monitor collects operating system statistics (system metrics) such as memory and swap space usage, processes, IO usage, and network related data. The Cluster Health Monitor collects information in real time and usually once a second. Cluster Health Monitor collects as much of system metrics and data as feasible that is restricted by the acceptable level of resource consumption by the tool.

Cluster Health Monitor provides system metrics and data for troubleshooting many different types of problems such as node reboot, instance eviction, server hang, severe performance degradation, and any other problems that need the system metrics and data. By monitoring the data constantly, users can use Cluster Health Monitor detect potential problem areas such as CPU load, memory constraints, and spinning processes before the problem causes an unwanted outage.

The information collected by Cluster Health Monitor is stored in the Grid Infrastructure Management Repository (GIMR). GIMR is a centralized infrastructure database for diagnostic and performance data and resides in the Grid home. It is a single instance multitenant database with a single pluggable database (PDB). It includes partitioning (for data lifecycle management). GIMR clients include: Cluster Health Monitor, Rapid Home Provisioning, Enterprise Manager Cloud Control, and Trace File Analyzer. Cluster Health Monitor puts all of its operating system metric data in GIMR. Rapid Home Provisioning uses GIMR to persist metadata about each of the database homes that it services. Enterprise Manager Cloud Control and Trace File Analyzer retrieve data from GIMR, predominantly Cluster Health Monitor data.

You can review the information collected by Cluster Health Monitor using either `diagcollection.pl` or `oclumon`:

- Diagcollection.pl: As the `grid` user, using command `*Grid_home*/bin/diagcollection.pl --collect --chmos` produces output for all the data collected in the repository. Because there might be a lot of data, this could take a long time. It is better to limit the query to an interesting time interval.

  ```
  Grid_home/bin/diagcollection.pl --collect --crshome Grid_home
    --chmos --incidenttime start_time --incidentduration 05:00
  ```

  The above command generates a report that covers 5 hours from the time specified by `incidenttime`. The `incidenttime` must be in the form `MM/DD/YYYYHH:MN:SS` where `MM` is month, `DD` is date, `YYYY` is year, `HH` is hour in 24 hour format, `MN` is minute, and `SS` is second. For example, if you want to put the incident time to start from 10:15 PM on June 01, 2016, the incident time is 06/01/201622:15:00. The `incidenttime` and `incidentduration` can be changed to capture more data.

- oclumon: Alternatively, you can use the command `oclumon dumpnodeview -allnodes -v -last "11:59:59" > *output-filename*`’ if `diagcollection.pl` fails with any reason. This command generates a report from the repository up to last 12 hours. The `-last` value can be changed to get more or less data.

10.1.4 Cluster Health Advisor

The Cluster Health Advisor daemon collects data from Oracle Real Application Clusters (Oracle RAC), Oracle RAC One Node databases, and the host operating system and hardware resources, to analyze and produce a root cause and action after finding a database or host performance issue.

The user can initiate a calibration process to generate data from which a predictive analysis data set is created. This baseline operation model is then evaluated every five seconds to determine whether an abnormal condition has occurred and is progressing, along with its root cause, details, and corrective actions. This data evidence persists in the Grid Infrastructure Management Repository database for reporting to Oracle Enterprise Manager Cloud Control.

The Cluster Health Advisor daemon provides system and database administrators with early warning of pending performance issues, and root causes and corrective actions for Oracle RAC databases and cluster nodes. This advanced proactive diagnostic capability will increase availability and performance management.

To track the online health of an Oracle RAC system, Cluster Health Advisor analyzes the system’s measurements (taken as samples of the operational state of a running system) to detect whether the measured values are within an established normal base line, or if they are drifting anomalously beyond their acceptable perimeter. If Cluster Health Advisor detects irregularities in any of the measured values, it invokes an incident diagnostic engine to determine the identity of the underlying issue and generates incident details. The analysis cycle repeats periodically with a fresh set of sampled measurements to track the health status of the system over time.

To view the findings of Cluster Health Advisor, you can use:here is a and a command-line interface. Graphical interfaces include:

- Graphical interface
  - Enterprise Manager Cloud Control
  - A standalone specialized console dedicated to depicting Cluster Health Advisor analysis
- Command line interface — CHACTL

Example 10-1 Using CHACTL to Review Cluster Health Advisor Findings

Here is an example of using CHACTL to review the findings of Cluster Health Advisor:

```
$ chactl query diagnosis –start start_time -end end_time
10:05:15          Database A        Disk Failure   (instance1) [detected]
                                      Storage Bandwidth Saturation (instance 1)
[detected]
10:05:30        Database A     Disk Failure (instance 1)  [cleared]
                               Storage Bandwidth Saturation (instance 2) [detected]
                               Storage Bandwidth Saturation (instance 1) [cleared]
                   Database B      Storage Bandwidth Saturation (instance 1 )
                   Host A              Disk Failure
```

10.1.5 Trace File Analyzer Collector

Trace File Analyzer centralizes and automates the collection of diagnostic information.

- About the Trace File Analyzer Collector (tafctl)
  Trace File Analyzer Collector (TFA) is a diagnostic collection utility to simplify diagnostic data collection on Oracle Grid Infrastructure, Oracle RAC and single-instance Oracle Database systems.
- tfactl Command Summary
- Using tfactl to Collect Diagnostic Logs for a Specific Time Period
  The diagcollection module of TFA has the capability to take a number of different parameters to allow the user to control how large or detailed the required evidence set is.

10.1.5.1 About the Trace File Analyzer Collector (tafctl)

Trace File Analyzer Collector (TFA) is a diagnostic collection utility to simplify diagnostic data collection on Oracle Grid Infrastructure, Oracle RAC and single-instance Oracle Database systems.

TFA is similar to the `diagcollection` utility packaged with Oracle Grid Infrastructure in that it collects and packages diagnostic data. However, TFA is much more powerful than `diagcollection` with its ability to centralize and automate the collection of diagnostic information.

TFA Collector Features include:

- Simplifies diagnostic data collections
- Performs cluster-wide diagnostic collection for all components with a single command
- Trims diagnostic files to include only information from around the incident time
- Consolidates collected diagnostics on a single node

TFA runs on each node of your cluster or on a single node where no Grid Infrastructure is installed. TFA consists of a daemon, and a command line interface (CLI). The TFA daemon is a Java Virtual Machine (JVM) that by default runs at all times on any configured node and can be identified as TFAMain. The CLI is made up of a Java command line processor that sends commands to the TFAMain secure socket and a perl script `tfactl`.

When TFA is installed with Oracle Grid Infrastructure, it is installed and configured through the installer, but can be patched through regular Patch Set Updates, or by downloading the latest TFA from My Oracle Support.

Upon installation completion a TFAMain JVM will be running on each node in the configuration and will run an inventory process to discover files in the trace directories found. The inventory process determines first and last timestamp as well as the file type for all files in these directories. Any alert type files discovered will then be monitored continuously for significant events and if any of those events occur then TFA can automatically gather relevant diagnostics if configured to do so, though manual collections may be initiated at any time. Alert type files are just CRS, ASM and RDBMS alert logs. Automatic diagnostic collections are disabled by default.

10.1.5.2 tfactl Command Summary

`tfactl` is a command-line interface for managing the Trace File Analyzer Collector (TFA) utility.

File Path

To Manage TFA, `tfactl` can be run from the *Oracle_base*/`bin` directory and can be run as the `root` user, or any other user authorized to run TFA. Non root users will only able to run commands that complete diagnostic collections or make directories they own available for collection. If user is given `sudo` access to run `tfactl` then they will have access to all commands.

Syntax

```
tfactl command [options]
```

tfactl Commands

Command	Description
`diagcollect`	Specify the type of information collected by TFA
`collection`	Manage TFA collections
`analyze`	Search for a pattern in the database or clusterware alert logs that occurred during a specified period of time in the past. Default time period is one hour.
`ips`	Create and manage the contents of a trace file package
`run`	Run the specified TFA support tool
`start` *`tool`*	Starts TFA. If a tool is specified, then it starts the specified tool.
`stop` *`tool`*	Stops TFA. If a tool is specified, then it stops the specified tool.
`enable`	Enable TFA autostart
`disable`	Disable TFA autostart
`status`	Check the run status of TFA process
`print`	Print requested details
`access`	Add or remove or list TFA users and groups
`purge`	Delete collections from the TFA repository
`directory`	Add or remove or modify the directory in TFA
`host`	Add or remove host in TFA
`set`	Turn on or turn off or modify various TFA features
`toolstatus`	Print the status of TFA support tools
`uninstall`	Uninstall TFA from the local node
`diagnosetfa`	Collect TFA diagnostics

See Also:

My Oracle Support note: TFA Collector - TFA with Database Support Tools Bundle (Doc ID 1513912.1) for a complete list of the options available for each command

10.1.5.3 Using tfactl to Collect Diagnostic Logs for a Specific Time Period

The diagcollection module of TFA has the capability to take a number of different parameters to allow the user to control how large or detailed the required evidence set is.

In this example, you use the `-from` and `-to` switches for the `tfactl diagcollect` command, which tells TFA to collect diagnostic logs of all types from midnight April 2nd midnight to 13:00 on April 2nd. The command will launch the specified diagnostic collection on all cluster nodes and place the results in a zip file for each node in the `$TFA_HOME/repository` directory on the node in which TFA was invoked.

```
[oracle@docrac1 ~]# ./tfactl diagcollect -from "Apr/02/2016" -to "Apr/02/2016
13:00:00"
Collecting data for all nodes
Scanning files from Apr/02/2016 00:00:00 to Apr/02/2016 13:00:00

Collection Id : 20160402160419racnode1

Repository Location in docrac1 : /u01/app/oracle/tfa/repository
2016/04/02 15:04:21 EDT : Collection Name : tfa_Thu_Apr_2_15_04_19_EDT_2016.zip
2016/04/02 15:04:21 EDT : Sending diagcollect request to host : cehaovmsp102
2016/04/02 15:04:22 EDT : Scanning of files for Collection in progress...
2016/04/02 15:04:22 EDT : Collecting extra files...
2016/04/02 15:04:32 EDT : Getting list of files satisfying time range [04/02/2014
00:00:00 EDT, 04/02/2014 13:00:00 EDT]
2016/04/02 15:04:32 EDT : Starting Thread to identify stored files to collect
2016/04/02 15:04:32 EDT : Getting List of Files to Collect
2016/04/02 15:04:32 EDT : Trimming file : docrac1/u01/app/11.2.0/grid/log/docrac1/
alertdocrac1.log with original file size : 9.7MB
2016/04/02 15:04:32 EDT : Trimming file : docrac1/u01/app/11.2.0/grid/log/docrac1/
client/oifcfg.log with original file size : 1.2MB
2016/04/02 15:04:32 EDT : Trimming file : docrac1/u01/app/11.2.0/grid/log/docrac1/
crfmond/crfmond.log with original file size : 3.4MB
2016/04/02 15:04:32 EDT : Trimming file : docrac1/u01/app/11.2.0/grid/log/docrac1/
evmd/evmd.log with original file size : 3.2MB
2016/04/02 15:04:32 EDT : Trimming file : docrac1/u01/app/11.2.0/grid/log/docrac1/
gpnpd/gpnpd.log with original file size : 608kB
2016/04/02 15:04:35 EDT : Trimming file : docrac1/rdbms/orcl/ORCL1/trace/
alert_ORCL1.log with original file size : 1.4MB
2016/04/02 15:04:35 EDT : Finshed Getting List of Files to Collect
2016/04/02 15:04:35 EDT : Trimming file : docrac1/tnslsnr/docrac1/listener_scan1/
trace/listener_scan1.log with original file size : 83MB
2016/04/02 15:04:35 EDT : Trimming file : docrac1/asm/+asm/+ASM1/trace/alert_
+ASM1.log with original file size : 623kB
2016/04/02 15:04:35 EDT : Collecting ADR incident files...
2016/04/02 15:04:35 EDT : Waiting for collection of extra files
2016/04/02 15:05:10 EDT : Completed collection of extra files...
2016/04/02 15:05:10 EDT : Completed Zipping of all files
2016/04/02 15:05:10 EDT : Cleaning up temporary files
2016/04/02 15:05:10 EDT : Finished Cleaning up temporary files
```

```
2016/04/02 15:05:10 EDT : Finalizing the Collection Zip File
2016/04/02 15:05:11 EDT : Finished Finalizing the Collection Zip File
2016/04/02 15:05:11 EDT : Total Number of Files checked : 17168
2016/04/02 15:05:11 EDT : Total Size of all Files Checked : 2.3GB
2016/04/02 15:05:11 EDT : Number of files containing required range : 29
2016/04/02 15:05:11 EDT : Total Size of Files containing required range : 104MB
2016/04/02 15:05:11 EDT : Number of files trimmed : 8
2016/04/02 15:05:11 EDT : Total Size of data prior to zip : 1.3MB
2016/04/02 15:05:11 EDT : Saved 103MB by trimming files
2016/04/02 15:05:11 EDT : Zip file size : 121kB
2016/04/02 15:05:11 EDT : Total time taken : 49s
2016/04/02 15:05:11 EDT : Remote Collection in Progress...
2016/04/02 15:05:22 EDT : cehaovmsp102:Completed Collection
2016/04/02 15:05:22 EDT : Completed collection of zip files.

Logs are being collected to: /u01/app/oracle/tfa/repository/
collection_Thu_Apr_2_15_04_19_EDT_2016_node_all
/u01/app/oracle/tfa/repository/collection_Thu_Apr_2_15_04_19_EDT_2016_node_all/
docrac1.tfa_Thu_Apr_2_15_04_19_EDT_2016.zip
/u01/app/oracle/tfa/repository/collection_Thu_Apr_2_15_04_19_EDT_2016_node_all/
cehaovmsp102.tfa_Thu_Apr_2_15_04_19_EDT_2016.zip
```

10.1.6 Hang Manager

Hang Manager can reliably detect and resolve hung database sessions in a timely manner.

Hang Manager is only active when Oracle RAC is enabled. You can determine whether an Oracle RAC database is used / enabled by querying the `CLUSTER_DATABASE` initialization parameter. If this parameter is set to `TRUE`, then Oracle RAC is enabled on this database.

Since Oracle Database 11*g* Release 2 (11.2.0.2), Hang Manager can resolve hangs it detects by terminating the root session or process. By default, Hang Manager will not terminate an instance or evict a node. Hang Manager does not resolve all hangs that it detects. For example, hangs which involve a possible application issue are left to the user to determine the correct course of action. Also, if Hang Manager determines that the instance on which the source of a hang resides is experiencing high CPU or IO load, then hang resolution is delayed. This allows the source session or process time to progress and for the hang to resolve on its own. Hang Manager only resolves hangs on Oracle Database instances; Hang Manager does not resolve any hangs on Oracle ASM instances.

Starting with Oracle Database 12*c* Release 1, if Oracle Database Quality of Service (QoS) Management is active on the cluster, then Hang Manager uses additional information provided by Oracle Database QoS Management to determine if a hang should be ignored or resolved.

When Hang Manager resolves a hang, it signals an `ORA-32701` incident which is visible in the alert log. The alert log message is similar to the following:

```
ORA-32701: Possible hangs up to hang ID=24 detected
Incident details in: /ee/oracle/oracle_base/diag/rdbms/orcl/orcl1/incident/
incdir_1944098/orcl1_dia0_34930694_i1944098.trc
DIA0 terminating blocker (ospid: 28311778 sid: 3398 ser#: 1) of hang with ID = 24
     requested by master DIA0 process on instance 2
     Hang Resolution Reason: Automatic hang resolution was performed to free a
    critical database process.
     by terminating session sid:3398 with serial # 1 (ospid:28311778)
```

The above message describes which hang is about to be resolved; the hang with ID 24. It also lists the root session ID, serial number and Operating System process ID and on which instance it resides. Finally, Hang Manager gives a short reason as to why this hang will be resolved. In this case, a regular user session is blocking a critical database process so the user session will be terminated to allow the critical database process to run.

> **Note:**
>
> An `ORA-32701` incident in the alert log does not indicate that there is a problem with Hang Manager. Instead, it indicates that Hang Manager has found a hang which will be resolved by terminating the root session.

You can monitor Hang Manager activity by querying the following data dictionary views:

- `V$HANG_INFO`: Contains the active hangs and deadlocks detected by Hang Manager. It does not include any hangs which have been resolved by Hang Manager or have self-resolved. Only the latest 32 active hangs are in this view.
- `V$HANG_SESSION_INFO`: Contains the sessions in the main chain for all of the hangs included in the `V$HANG_INFO` view. Only the first 20 sessions of the main chain, including the root of the hang, for each hang are kept in this view.
- `GV$HANG_STATISTICS` or `V$HANG_STATISTICS`: Contains various Hang Manager statistics regarding hangs or deadlocks which it has detected. These statistics include: the number of hangs detected, the number of deadlocks detected, the number of hangs resolved by Hang Manager, the number of hangs ignored broken down by the reason for ignoring the hang, the number of hangs which self-resolved, and so on.

10.1.7 Managing Memory Pressure for Database Servers

Memory Guard detects memory pressure on a server in real time and redirects new sessions to other servers to prevent using all available memory on the stressed server.

Enterprise database servers can use all available memory due to too many open sessions or runaway workloads. Running out of memory can result in failed transactions or, in extreme cases, a restart of the server and the loss of a valuable resource for your applications.

Rerouting new sessions to different servers protects the existing workloads on the memory-stressed server and enables the server to remain available. Memory Guard is a feature of Oracle RAC that manages the memory pressure for servers, adding a new resource protection capability in managing service levels for applications hosted on Oracle RAC databases.

When Oracle Database Quality of Service Management is enabled and managing an Oracle Clusterware server pool, Cluster Health Monitor sends a metrics stream that provides real-time information about memory resources for the cluster servers to Memory Guard. This information includes:

- Amount of available memory
- Amount of memory currently in use

If Memory Guard determines that a node is experiencing memory pressure, then the database services managed by Oracle Clusterware are stopped on that node, preventing new connections from being created. After the memory stress is relieved, the services on that node are restarted automatically, and the listener starts sending new connections to that server. The memory pressure can be relieved in several ways (for example, by closing existing sessions or by user intervention).

10.1.8 Oracle Database Quality of Service Management

Oracle Database Quality of Service (QoS) Management is an automated, policy-based product that monitors the workload requests for an entire system.

Oracle Database QoS Management manages the resources that are shared across applications, and adjusts the system configuration to keep the applications running at the performance levels needed by your business. Oracle Database QoS Management responds gracefully to changes in system configuration and demand, thus avoiding additional oscillations in the performance levels of your applications.

Oracle Database QoS Management monitors the performance of each work request on a target system. Oracle Database QoS Management starts to track a work request from the time a work request requests a connection to the database using a database service. The amount of time required to complete a work request, or the response time (also known as the end-to-end response time, or round-trip time), is the time from when the request for data was initiated and when the data request is completed. By accurately measuring the two components of response time, which are the time spent using resources and the time spent waiting to use resources, Oracle Database QoS Management can quickly detect bottlenecks in the system. Oracle Database QoS Management then makes suggestions to reallocate resources to relieve a bottleneck, thus preserving or restoring service levels.

Oracle Database QoS Management manages the resources on your system so that:

- When sufficient resources are available to meet the demand, business-level performance requirements for your applications are met, even if the workload changes.
- When sufficient resources are *not* available to meet the demand, Oracle Database QoS Management attempts to satisfy the more critical business performance requirements at the expense of less critical performance requirements.

Benefits of Using Oracle Database QoS Management

In a typical company, when the response times of your applications are not within acceptable levels, problem resolution can be very slow. Often, the first questions that administrators ask are: "Did we configure the system correctly? Is there a parameter change that fixes the problem? Do we need more hardware?" Unfortunately, these questions are very difficult to answer precisely. The result is often hours of unproductive and frustrating experimentation.

Oracle Database QoS Management provides the following benefits:

- Reduces the time and expertise requirements for system administrators who manage Oracle Real Application Clusters (Oracle RAC) resources.
- Helps reduce the number of performance outages.
- Reduces the time needed to resolve problems that limit or decrease the performance of your applications.

- Provides stability to the system as the workloads change.
- Makes the addition or removal of servers transparent to applications.
- Reduces the impact on the system caused by server failures.
- Helps ensure that service-level agreements (SLAs) are met.
- Enables more effective sharing of hardware resources.

Oracle Database QoS Management helps manage the resources that are shared by applications in a cluster. Oracle Database QoS Management can help identify and resolve performance bottlenecks. Oracle Database QoS Management does not diagnose or tune application or database performance issues. When tuning the performance of your applications, the goal is to achieve optimal performance. Oracle Database QoS Management does not seek to make your applications run faster, but instead works to remove obstacles that prevent your applications from running at their optimal performance levels.

> **See Also:**
>
> *Oracle Database Quality of Service Management User's Guide*

10.2 Monitoring Oracle Clusterware Using Enterprise Manager

You can use Enterprise Manager to monitor Oracle Clusterware.

Other monitoring features available with Enterprise Manager include:

- Viewing the status of Oracle Clusterware on each node of the cluster
- Receiving notifications if there are any VIP relocations
- Monitoring the overall throughput across the private interconnect
- Receiving notifications if nodeapps go down or come up
- Viewing alerts if a database instance is using the public interface instead of the VIP
- Monitoring the Clusterware alert log for OCR or voting disk related issues, node evictions, and other clusterware errors

You can also monitor Oracle Clusterware resources and use cluster component health monitoring for individual components. To monitor resources you click the **Administration** link from the Cluster home page. To monitor the health of individual cluster components, click the **All Metrics** link in the Related Links section of the Cluster home page.

- Accessing the Oracle Clusterware Information
  From the Cluster Database Home page, there a several ways to access Oracle Clusterware information.
- Viewing the Cluster Topology Page
  The Oracle Enterprise Manager Topology Viewer enables you to visually see the relationships between target types in your cluster.

- Viewing Cluster Health Monitor Data from Enterprise Manager
  You can use Oracle Enterprise Manager Cloud Control to view data from Cluster Health Monitor (CHM).

> **See Also:**
>
> - "About Verifying the Oracle Clusterware Installation"
> - "Using CRSCTL to Diagnose Cluster Issues"

10.2.1 Accessing the Oracle Clusterware Information

From the Cluster Database Home page, there a several ways to access Oracle Clusterware information.

To access Oracle Clusterware information:

1. Access the Cluster Database Home page of Oracle Enterprise Manager.

   See *Oracle Database 2 Day DBA* for details on logging in to Enterprise Manager.
2. In the General section, click the link next to **Cluster** to view the Cluster Home page.
3. On the Cluster Home page, in the Diagnostic Summary section, click the number next to **Interconnect Alerts** to view the Interconnects subpage for the cluster.

   Alternatively, from the Cluster menu, select **Interconnects** to display the Interconnects page.
4. From the Cluster menu, select **Configuration**, and then select **Topology**. Click a node in the graphical display to activate the controls.
5. Click the **Interface** component. Right-click the Interface component, then choose **View Details** from the menu to display the Interconnects page for the cluster.
6. In the Cluster Databases section of the Cluster Home page, select the name of a cluster database to return to the Cluster Database Home page for that database.

> **See Also:**
>
> "Viewing the Cluster Topology Page"

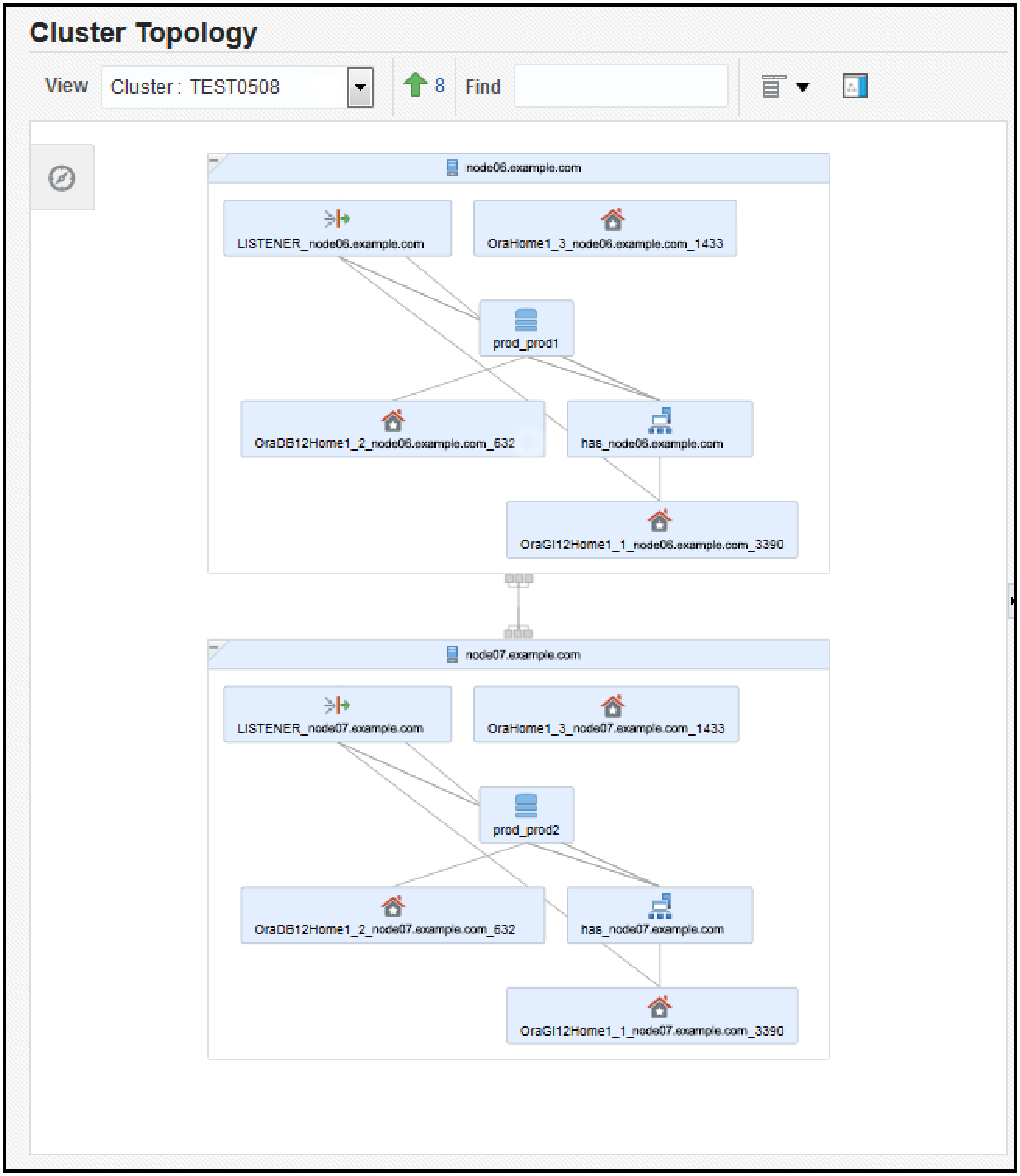
10.2.2 Viewing the Cluster Topology Page

The Oracle Enterprise Manager Topology Viewer enables you to visually see the relationships between target types in your cluster.

You can zoom in or out, pan, and see selection details. The Topology Viewer uses individually distinct icons to represent system target types, and standardized visual indicators, such as frames around selections, are used across all target types.

The Topology Viewer populates icons based on your system configuration. If a listener is serving an instance, then a line connects the listener icon and the instance icon. If a cluster database is configured to use Oracle ASM, then the relationship between the cluster Oracle ASM and cluster database appears in the topology, as shown in the following figure:

Figure 10-1 Cluster Topology Page of Oracle Enterprise Manager


Cluster Topology
View
Cluster : TEST0508
8
Find
node06.example.com
LISTENER_node06.example.com
OraHome1_3_node06.example.com_1433
prod_prod1
OraDB12Home1_2_node06.example.com_632
has_node06.example.com
OraGI12Home1_1_node06.example.com_3390
node07.example.com
LISTENER_node07.example.com
OraHome1_3_node07.example.com_1433
prod_prod2
OraDB12Home1_2_node07.example.com_632
has_node07.example.com
OraGI12Home1_1_node07.example.com_3390

If the **Show Configuration Details** option is unchecked, then the topology shows the monitoring view of the environment, which includes general information such as alerts and overall status. If you select the **Show Configuration Details** option, then additional details are shown in the Selection Details page, which are valid for any topology view. For instance, the listener component would also show the computer name and port number.

You can click an icon and then the right mouse button to display a menu of available actions. Several actions go to pages related to the target type where you can perform monitoring or tuning tasks.

Refer to the Enterprise Manager Online Help for more information about the contents of this page.

10.2.3 Viewing Cluster Health Monitor Data from Enterprise Manager

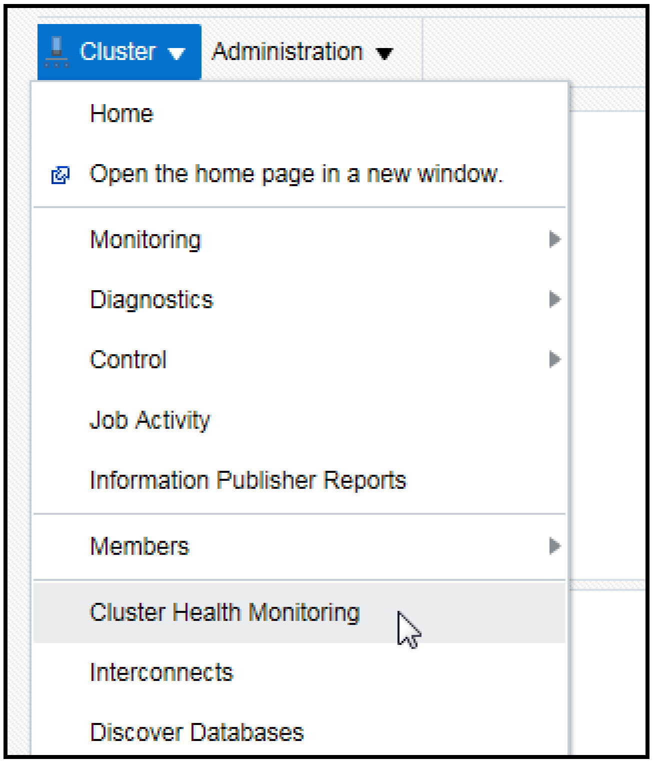
You can use Oracle Enterprise Manager Cloud Control to view data from Cluster Health Monitor (CHM).

The metric data from Cluster Health Monitor is available in graphical display within Enterprise Manager Cloud Control. Complete cluster views of this data are accessible from the cluster target page. You can view CHM data for the last day in overview format for the entire cluster. The metric categories displayed in the graphs are CPU, Memory and Network.

Each category can be separately displayed in greater detail, showing additional metrics. For example, selecting CPU will result in cluster graphs detailing CPU System Usage, CPU User Usage, and CPU Queue length. From any cluster view you can select individual node views to more closely examine performance of a single server. As in the case of CPU, the performance of each core will be displayed. You can at any time move your cursor along the graph to see a tool-tip displaying the numerical values and time stamp of that point.

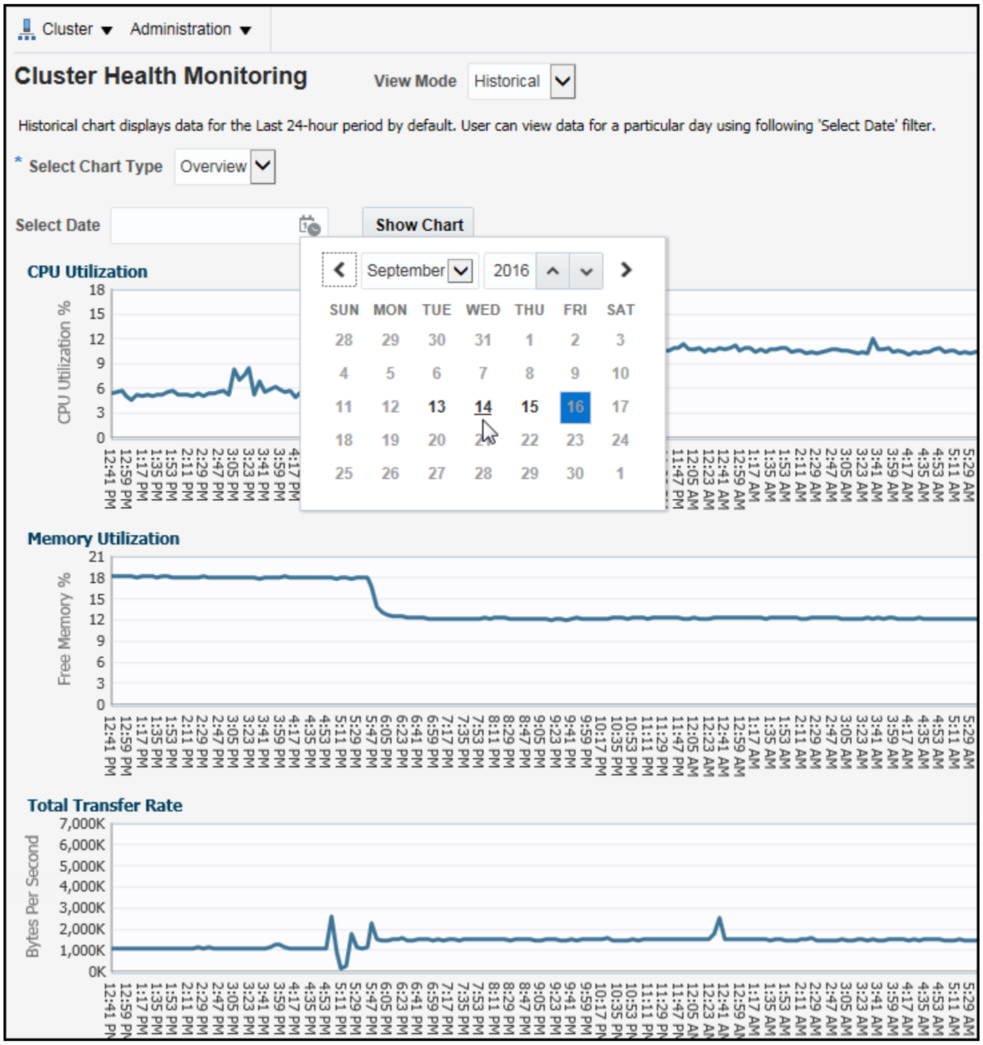
Besides examining the current day’s performance, you can also review historical data. The amount of historical data is governed by the retention time configured for CHM repository in the Gird Infrastructure Management Repository. The default is 72 hours. You can view the available historical data by using the **View Mode** drop-down menu and selecting **Historical**. A previous date can then be entered or selected from a pop-up calendar that displays the dates where data is available in bold font. Selecting **Show Chart** will then display the selected metrics graphs.

1. Select the target cluster in Enterprise Manager.
2. From the **Cluster** menu, select **Cluster Health Monitoring**.



The Cluster Health Monitoring page appears.

3. View the graphs. To view data from the past, change the View Mode from Real Time to Historical, and select the date to view.



Cluster ▾ Administration ▾

Cluster Health Monitoring

View Mode: Historical

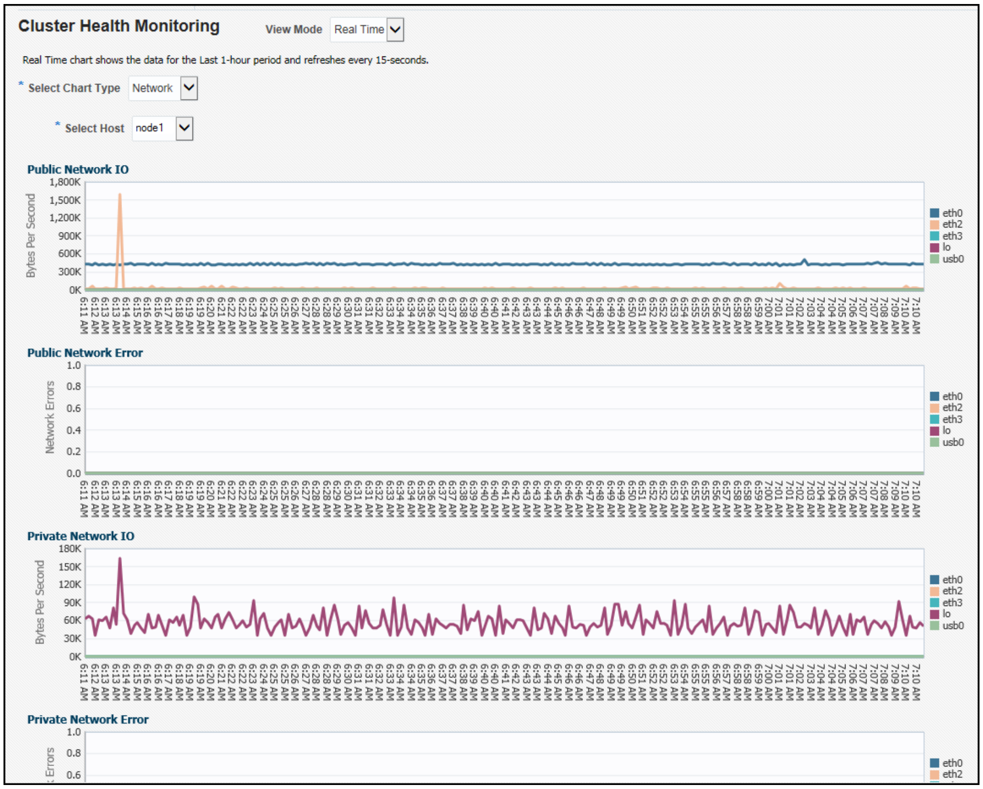
Historical chart displays data for the Last 24-hour period by default. User can view data for a particular day using following 'Select Date' filter.

* Select Chart Type: Overview

Select Date

Show Chart
September 2016
SUN MON TUE WED THU FRI SAT
28 29 30 31 1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 1
CPU Utilization
CPU Utilization %
0 3 6 9 12 15 18
Memory Utilization
Free Memory %
0 3 6 9 12 15 18 21
Total Transfer Rate
Bytes Per Second
0K 1,000K 2,000K 3,000K 4,000K 5,000K 6,000K 7,000K
12:41 PM 12:59 PM 1:17 PM 1:35 PM 1:53 PM 2:11 PM 2:29 PM 2:47 PM 3:05 PM 3:23 PM 3:41 PM 3:59 PM 4:17 PM 4:35 PM 4:53 PM 5:11 PM 5:29 PM 5:47 PM 6:05 PM 6:23 PM 6:41 PM 6:59 PM 7:17 PM 7:35 PM 7:53 PM 8:11 PM 8:29 PM 8:47 PM 9:05 PM 9:23 PM 9:41 PM 9:59 PM 10:17 PM 10:35 PM 10:53 PM 11:11 PM 11:29 PM 11:47 PM 12:05 AM 12:23 AM 12:41 AM 12:59 AM 1:17 AM 1:35 AM 1:53 AM 2:11 AM 2:29 AM 2:47 AM 3:05 AM 3:23 AM 3:41 AM 3:59 AM 4:17 AM 4:35 AM 4:53 AM 5:11 AM 5:29 AM

4. If you change the Chart Type to CPU, Memory, or Network, you can view additional details, such as CPU User Usage, Private Network IO, or Swap Free Memory Percentage.
5. While viewing the Network or CPU metric graphs, you can click on a node name to view additional details just for that node, such as the Public Network IO traffic by network interface, or the CPU usage for each core.



Cluster Health Monitoring

View Mode: Real Time

Real Time chart shows the data for the Last 1-hour period and refreshes every 15-seconds.

* Select Chart Type: Network

* Select Host: node1

10.3 Troubleshooting Configuration Problems in Oracle RAC Environments

The Cluster Ready Services Control (CRSCTL) command `check` enables you to determine the status of several Oracle Clusterware components simultaneously.

Problems can occur when attempting to complete the installation or database creation process manually instead of using the Oracle Database management tools. Other problems occur due to the database administrator or system administrator missing important operating system or cluster configuration steps before installation. Both Oracle Clusterware and Oracle Database components have subcomponents that you can troubleshoot.

- About the Oracle Clusterware Alert Log
  Oracle Clusterware posts alert messages when important events occur. For example, you might see alert messages from the Cluster Ready Services (CRS) daemon process when it starts, if it aborts, if the failover process fails, or if automatic restart of an Oracle Clusterware resource failed.
- About the Oracle Clusterware Component Log Files
  Starting with Oracle Clusterware 12*c* release 1 (12.1.0.2), diagnostic data files written by Oracle Clusterware programs are known as trace files and have a `.trc` file extension, and appear together in the `trace` subdirectory of the Automatic Diagnostic Repository (ADR) home.

- Using CRSCTL to Diagnose Cluster Issues
  You can use CRSCTL commands as the `root` operating system user to diagnose problems with your Oracle Clusterware installation, or to enable dynamic debugging for Oracle Clusterware.
- Viewing Oracle RAC Database Alert Log Messages
  An alert log is created for each instance in a cluster database.

> **See Also:**
>
> - "Tools for Installing, Configuring, and Managing Oracle RAC"
> - "About Verifying the Oracle Clusterware Installation"

10.3.1 About the Oracle Clusterware Alert Log

Oracle Clusterware posts alert messages when important events occur. For example, you might see alert messages from the Cluster Ready Services (CRS) daemon process when it starts, if it aborts, if the failover process fails, or if automatic restart of an Oracle Clusterware resource failed.

Enterprise Manager monitors the Clusterware log file and posts an alert on the Cluster Home page if an error is detected. For example, if a voting disk is not available, then a `CRS-1604` error is raised, and a critical alert is posted on the Cluster Home page. You can customize the error detection and alert settings on the Metric and Policy Settings page.

The Oracle Clusterware alert log should be the first place to look for serious errors. It often contains references to other diagnostic logs that provide detailed information on a specific component. The location of the Oracle Clusterware log file is `ORACLE_BASE/diag/crs/`*`hostname`*`/crs/trace/alert.log`, where `ORACLE_BASE` is the Oracle base path you specified when you installed Oracle Grid Infrastructure and *`hostname`* is the name of the host.

> **See Also:**
>
> - *Oracle Clusterware Administration and Deployment Guide* for more information about Oracle Clusterware diagnostic and alert log data
> - "Monitoring Oracle Clusterware Using Enterprise Manager"

10.3.2 About the Oracle Clusterware Component Log Files

Starting with Oracle Clusterware 12*c* release 1 (12.1.0.2), diagnostic data files written by Oracle Clusterware programs are known as trace files and have a `.trc` file extension, and appear together in the `trace` subdirectory of the Automatic Diagnostic Repository (ADR) home.

Besides trace files, the `trace` subdirectory in the Oracle Clusterware ADR home contains the simple text Oracle Clusterware alert log. It always has the name alert.log. The alert log is also written as an XML file in the alert subdirectory of the ADR home, but the text alert log is most easily read.

> **See Also:**
>
> *Oracle Clusterware Administration and Deployment Guide*

10.3.3 Using CRSCTL to Diagnose Cluster Issues

You can use CRSCTL commands as the root operating system user to diagnose problems with your Oracle Clusterware installation, or to enable dynamic debugging for Oracle Clusterware.

- Checking the Status of the Oracle Clusterware Installation
  Use the CRSCTL check command to display the status of an Oracle Clusterware component or daemon.
- Enabling Debugging of Oracle Clusterware Components
  You can enable debugging for the Oracle Cluster daemons, Event Manager (EVM), and their modules by running CRSCTL commands.
- Enabling Debugging for an Oracle Clusterware Resource
  You can use CRSCTL commands to enable debugging for resource managed by Oracle Clusterware.
- Enabling Oracle Clusterware Daemons
  When the Oracle Clusterware daemons are enabled, they start automatically when the node is started.
- Disabling Oracle Clusterware Daemons
  To prevent the daemons from starting automatically, you can disable them using crsctl commands.

> **See Also:**
>
> - "Tools for Installing, Configuring, and Managing Oracle RAC"
> - *Oracle Clusterware Administration and Deployment Guide*

10.3.3.1 Checking the Status of the Oracle Clusterware Installation

Use the CRSCTL check command to display the status of an Oracle Clusterware component or daemon.

To determine the condition of your Oracle Clusterware installation:

1. Log in to the operating system as the root user in a command window.
2. Use CRSCTL to check the status of Oracle Clusterware using the following command:

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

3. Check the status of an individual Oracle Clusterware daemon using the following syntax, where *daemon* is crsd, cssd, or evmd:

   ```
   # crsctl check daemon
   ```

4. To list the status of all Oracle Clusterware resources running on any node in the cluster, use the following command:

   ```
   # crsctl status resource -t
   ```

   This command lists the status of all registered Oracle Clusterware resources, which includes the VIPs, listeners, databases, services, and Oracle ASM instances and disk groups in a tabular form.

> **See Also:**
>
> crsctl check cluster

10.3.3.2 Enabling Debugging of Oracle Clusterware Components

You can enable debugging for the Oracle Cluster daemons, Event Manager (EVM), and their modules by running CRSCTL commands.

To enable debugging of Oracle Clusterware components:

1. In a command window, log in to the operating system as the root user.
2. Use the following command to obtain the module names for a component, where *component_name* is crs, evm, css or the name of the component for which you want to enable debugging:

   ```
   # crsctl lsmodules component_name
   ```

   For example, viewing the modules of the css component might return the following results:

   ```
   # crsctl lsmodules css
   The following are the CSS modules ::
   CSSD
   COMMCRS
   COMMNS
   ```

3. Use CRSCTL as follows, where *component_name* is the name of the Oracle Clusterware component for which you want to enable debugging, *module* is the name of module, and *debugging_level* is a number from 1 to 5:

   ```
   # crsctl set log component module:debugging_level
   ```

   For example, to enable the lowest level of tracing for the CSSD module of the css component, you would use the following command:

   ```
   # crsctl set log css CSSD:1
   ```

See Also:

Oracle Clusterware Administration and Deployment Guide

10.3.3.3 Enabling Debugging for an Oracle Clusterware Resource

You can use CRSCTL commands to enable debugging for resource managed by Oracle Clusterware.

To enable debugging of an Oracle Clusterware resource:

1. In a command window, log in to the operating system as the `root` user.
2. Obtain a list of the resources available for debugging by running the following command:

   ```
   # crsctl check crs
   ```

3. Run the following command to enable debugging, where *`resource_name`* is the name of an Oracle Clusterware resource, such as `ora.racnode1.vip`, and *`debugging_level`* is a number from 1 to 5:

   ```
   # crsctl set log crs "resource_name:debugging_level"
   ```

See Also:

Oracle Clusterware Administration and Deployment Guide

10.3.3.4 Enabling Oracle Clusterware Daemons

When the Oracle Clusterware daemons are enabled, they start automatically when the node is started.

To enable automatic startup for all Oracle Clusterware daemons:

1. In a command window, log in to the operating system as the `root` user.
2. Run the following CRSCTL command:

   ```
   # crsctl enable crs
   ```

See Also:

`crsctl enable crs`

10.3.3.5 Disabling Oracle Clusterware Daemons

To prevent the daemons from starting automatically, you can disable them using `crsctl` commands.

To disable automatic startup for all Oracle Clusterware daemons:

1. In a command window, log in to the operating system as the `root` user.
2. Run the following CRSCTL command:

   ```
   # crsctl disable crs
   ```

> **See Also:**
>
> `crsctl disable crs`

10.3.4 Viewing Oracle RAC Database Alert Log Messages

An alert log is created for each instance in a cluster database.

To view the alert log for an Oracle RAC database instance:

1. On the Cluster Database Home page, click the link for **Instances**.
2. Click the name of the instance for which you want to view the alert log.

   The Cluster Database Instance Home page appears.
3. From the Oracle Database menu, select **Logs**, then select **Alert Log Errors**.

   The Alert Log Errors page appears.
4. Optional: Click **XML Alert Log Contents** in the Related Links section to view all the entries in the alert log.

   On the View Alert Log Contents page, click **Go** to view the most recent entries, or you can enter your own search criteria.

> **See Also:**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for more information about where to find files for analyzing errors in Oracle RAC

11 Adding and Deleting Nodes and Instances

You can extend an existing Oracle Real Application Clusters (Oracle RAC) home to other nodes and instances in the cluster, and delete Oracle RAC instances and remove nodes from the cluster.

You can add nodes when configuring a new Oracle RAC cluster, or you can remove nodes to resize an existing cluster.

> **Note:**
>
> When adding or deleting nodes, it is very important that you perform each step in the order shown.

- Preparing the New Node
  Before a node can be added to the cluster, you must perform the same preinstallation steps on the new node as you did for all the existing nodes in the cluster.
- Verifying the New Node Meets the Prerequisites for Installation
  When adding a node to an existing cluster, the new node must match the configuration of the other nodes in the cluster.
- Extending the Oracle Grid Infrastructure Home to the New Node
  After the new node has been configured to support Oracle Clusterware, you use Oracle Universal Installer (OUI) to add a Grid home to the node being added to your cluster.
- Extending the Oracle RAC Home Directory
  Now that you have extended the Grid home to the new node, you must extend the Oracle home on `racnode1` to `racnode3`.
- Adding the New Node to the Cluster using Enterprise Manager
  The newly added node is automatically detected by Oracle Enterprise Manager.
- Creating an Instance on the New Node
  You can add an instance to the cluster using either the Instance Management option of Database Configuration Assistant (DBCA) or using Enterprise Manager.
- Deleting an Instance From the Cluster Database
  You can delete an instance from the cluster using either the Instance Management option of Database Configuration Assistant (DBCA) or using Enterprise Manager.
- Removing a Node From the Cluster
  Removing a node from the cluster can be as easy as simply shutting down the server.

> **See Also:**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

11.1 Preparing the New Node

Before a node can be added to the cluster, you must perform the same preinstallation steps on the new node as you did for all the existing nodes in the cluster.

This includes the following tasks:

- Checking hardware compatibility
- Configuring the operating system
- Configuring SSH connectivity between the new node and the other cluster members
- Configuring access to shared storage
- Creating groups, users, and directories

> **See Also:**
>
> - " Preparing Your Cluster "
> - "Configuring the Network"
> - "About Verifying the Oracle Clusterware Installation"

11.2 Verifying the New Node Meets the Prerequisites for Installation

When adding a node to an existing cluster, the new node must match the configuration of the other nodes in the cluster.

The new nodes must run the same operating system and version of the operating system as the other nodes. The new computer must also use the same chip architecture (32-bit or 64-bit) as the other nodes. However, you can have computers of different speeds and sizes in the same cluster.

After you have configured the new nodes, use `cluvfy` to verify that all the requirements for installation have been met. To verify the new node meets the hardware requirement, run the following command on an existing node (for example, either `racnode1` or `racnode2`) from the *`Grid_home`*`/bin` directory:

```
cluvfy stage -pre crsinst -n racnode3 -verbose
```

See Also:

- " Preparing Your Cluster "
- "Configuring the Network"
- "About Verifying the Oracle Clusterware Installation"

11.3 Extending the Oracle Grid Infrastructure Home to the New Node

After the new node has been configured to support Oracle Clusterware, you use Oracle Universal Installer (OUI) to add a Grid home to the node being added to your cluster.

This task assumes that you are adding a node named `racnode3` and that you have successfully installed Oracle Clusterware on `racnode1` in a nonshared home, where *Grid_home* represents the successfully installed Oracle Clusterware home.

To extend the Oracle Grid Infrastructure for a cluster home to include the new node:

1. Verify the new node has been properly prepared for an Oracle Clusterware installation by running the following CLUVFY command on the `racnode1` node:

   ```
   cluvfy stage -pre nodeadd -n racnode3 -verbose
   ```

2. As the `grid` user (owner of the software installation for Oracle Grid Infrastructure for a cluster) on `racnode1`, go to `*Grid_home*/addnode` and run the `addNode.sh` script in silent mode:

   - If you are using Grid Naming Service (GNS):

     ```
     ./addNode.sh -silent "CLUSTER_NEW_NODES={racnode3}"
     ```

   - If you are not using Grid Naming Service (GNS):

     ```
     ./addNode.sh -silent "CLUSTER_NEW_NODES={racnode3}" "CLUSTER_NEW_VIRTUAL_
     HOSTNAMES={racnode3-vip}"
     ```

   When running these commands, the curly braces (`{ }`) are not optional and must be included or the command returns an error.

   You can alternatively use a response file instead of placing all the arguments in the command line. See *Oracle Clusterware Administration and Deployment Guide* for more information on using response files.

3. When the script finishes, run the `root.sh` script as the `root` user on the new node, `racnode3`, from the Oracle home directory on that node.

4. If you are not using Oracle Grid Naming Service (GNS), then you must add the name and address for `racnode3` to DNS.

You should now have Oracle Clusterware running on the new node. To verify the installation of Oracle Clusterware on the new node, you can run the following command on the newly configured node, `racnode3`:

```
$ cd /u01/app/12.2.0/grid/bin
$ ./cluvfy stage -post nodeadd -n racnode3 -verbose
```

> **Note:**
>
> Avoid changing host names after you complete the Oracle Clusterware installation, including adding or deleting domain qualifications. Nodes with changed host names must be deleted from the cluster and added back with the new name.

> **See Also:**
>
> - "Completing the Oracle Clusterware Configuration"
> - *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database
> - *Oracle Clusterware Administration and Deployment Guide* for more information on using response files

11.4 Extending the Oracle RAC Home Directory

Now that you have extended the Grid home to the new node, you must extend the Oracle home on racnode1 to racnode3.

The procedure for adding an Oracle home to the new node is very similar to the procedure you just completed for extending the Grid home to the new node.

The following steps assume that you have completed the tasks described in the previous sections, "Preparing the New Node" and "Extending the Oracle Grid Infrastructure Home to the New Node", and that racnode3 is a member node of the cluster to which racnode1 belongs.

To extend the Oracle RAC installation to include the new node:

1. Ensure that you have successfully installed the Oracle RAC software on at least one node in your cluster environment. To use these procedures as shown, replace *Oracle_home* with the location of your installed Oracle home directory.
2. Go to the *Oracle_home/*addnode directory on racnode1 and run the addNode.sh script in silent mode as shown in the following example:

   ```
   $ cd /u01/app/oracle/product/12.2.0/dbhome_1/addnode
   $ ./addNode.sh -silent "CLUSTER_NEW_NODES={racnode3}"
   ```

3. When the script finishes, run the root.sh script as the root user on the new node, racnode3, from the Oracle home directory on that node.

   For policy-managed databases with Oracle Managed Files (OMF) enabled, no further actions are needed.

For a policy-managed database, when you add a new node to the cluster, it is placed in the Free pool by default. If you increase the cardinality of the database server pool, then an Oracle RAC instance is added to the new node, `racnode3`, and it is moved to the database server pool. No further action is necessary.

4. Add shared storage for the undo tablespace and redo log files.

   If OMF is not enabled for your database, then you must manually add an undo tablespace and redo logs.

5. If you have an administrator-managed database, then add a new instance on the new node as described in "Creating an Instance on the New Node".

   If you followed the installation instructions in this guide, then your cluster database is an administrator-managed database and stores the database files on Oracle Automatic Storage Management (Oracle ASM) with OMF enabled.

After completing these steps, you should have an installed Oracle home on the new node.

See Also:

- "Verifying Your Oracle RAC Database Installation"
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

11.5 Adding the New Node to the Cluster using Enterprise Manager

The newly added node is automatically detected by Oracle Enterprise Manager.

If you followed the steps described in "Extending the Oracle RAC Home Directory", then the cluster node is added to the cluster by the `addNode.sh` script. After the software is started on the new node, it is detected by Oracle Enterprise Manager. If an Oracle Enterprise Manager agent was not installed on the new node, then an alert is issued for that host, with the message "Incomplete configuration."

See Also:

- "Completing the Oracle Clusterware Configuration"
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

11.6 Creating an Instance on the New Node

You can add an instance to the cluster using either the Instance Management option of Database Configuration Assistant (DBCA) or using Enterprise Manager.

Before using either of these options you must first configure the new node to be a part of the cluster and install the software on the new node as described in the previous sections.

> **Note:**
>
> The steps described in this section require a license for the Enterprise Manager Provisioning Management pack. Refer to the *Oracle Database Licensing Information* for information about the availability of these features on your system.

There are two methods of adding an instance to the new node:

- Adding a New Instance for a Policy-Managed Database
  To add an instance to a policy-managed database, you simply increase the cardinality of the server pool for the database.
- Adding a New Instance for an Administrator-Managed Database
  When adding an instance to an administrator-managed database, you must specify the name of the database instance and which node it should run on.

> **See Also:**
>
> "About Oracle Real Application Clusters"

11.6.1 Adding a New Instance for a Policy-Managed Database

To add an instance to a policy-managed database, you simply increase the cardinality of the server pool for the database.

The database instance and Oracle ASM instance on the new node are created and configured automatically when a node is added to the server pool.

To add an instance to a policy-managed database using Enterprise Manager:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.
2. From the **Availability** menu, select **Cluster Database Operations**, and then select **Add Instance**.

   The Add Instance: Cluster Credentials page appears.

3. Enter the credentials of the Oracle Clusterware software owner and the SYSASM user, then click **Next**.

   After the credentials have been validated, the Edit Server Pool page appears.

4. Modify the server pool settings then click **Submit**.
5. From the **Cluster Database** menu, select **Home** to return to the Cluster Database Home page.
6. Review the number of instances available for your cluster database.

> **See Also:**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

11.6.2 Adding a New Instance for an Administrator-Managed Database

When adding an instance to an administrator-managed database, you must specify the name of the database instance and which node it should run on.

To add an instance to an administrator-managed database using Enterprise Manager:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Availability** menu, select **Cluster Database Operations**, and then select **Add Instance**.

   The Add Instance: Cluster Credentials page appears.

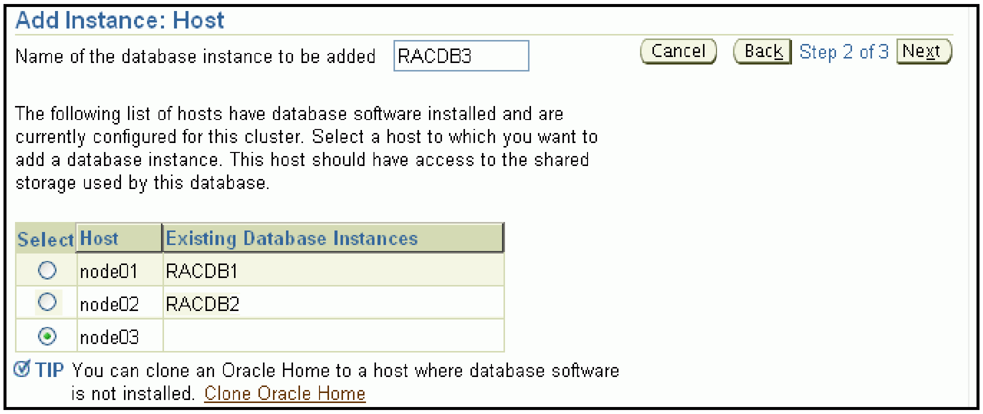
3. Enter the credentials of the Oracle Clusterware software owner and for the SYSASM user, then click **Next**.

   After the credentials have been validated, the Add Instance: Host page appears.

4. In the **Name of the database instance to be added** field, either use the default instance name, or enter a different name for the database instance, such as `racnode3`.
5. Select the node on which you want to create the new instance, then **Next**.

   > **Note:**
   >
   > This procedure assumes that the Oracle Database software is configured on the selected node and that there is no instance for the cluster database currently running on the selected node.

Figure 11-1 Add Instance: Host Page of Oracle Enterprise Manager



Add Instance: Host

Name of the database instance to be added: RACDB3

The following list of hosts have database software installed and are currently configured for this cluster. Select a host to which you want to add a database instance. This host should have access to the shared storage used by this database.

Select	Host	Existing Database Instances
○	node01	RACDB1
○	node02	RACDB2
●	node03	

TIP You can clone an Oracle Home to a host where database software is not installed. Clone Oracle Home

After the selected host has been validated, the Add Instance: Review page appears.

6. Review the information, then click **Submit Job** to proceed.

   A confirmation page appears.

7. Click **View Job** to check on the status of the submitted job.

   The Job Run detail page appears.

8. Click your browser's Refresh button until the job shows a status of Succeeded or Failed.

   If the job shows a status of Failed, then you can click the name of the step that failed to view the reason for the failure.

9. From the **Cluster Database** menu, select **Home** to return to the Cluster Database Home page.

   The number of instances available in the cluster database is increased by one.

> **See Also:**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

11.7 Deleting an Instance From the Cluster Database

You can delete an instance from the cluster using either the Instance Management option of Database Configuration Assistant (DBCA) or using Enterprise Manager.

Sometimes, it might be necessary to remove a database instance from your cluster database. For example, to retire or repurpose a server, you first remove the database instance running on that server.

> **Note:**
>
> The steps described in this section require a license for the Enterprise Manager Provisioning Management pack. Refer to the *Oracle Database Licensing Information* for information about the availability of these features on your system.

If you choose to use DBCA to delete an instance, then start the DBCA utility from a node that will remain part of the cluster.

When you delete an instance from a clustered database, there are two methods:

- Deleting an Instance From a Policy-Managed Database
  To delete an instance from a policy-managed databases, you simply decrease the cardinality of the server pool for the database.
- Deleting an Instance From an Administrator-Managed Database
  To remove an instance from an administrator-managed database, you must specify the name of the database instance to delete and which node the instance is running on.

> **See Also:**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for more information about removing Oracle RAC from a cluster node

11.7.1 Deleting an Instance From a Policy-Managed Database

To delete an instance from a policy-managed databases, you simply decrease the cardinality of the server pool for the database.

The database instance and Oracle ASM instance on the deallocated node are removed from the cluster and the node is either reassigned to another server pool or placed in the Free pool.

To delete a policy-managed database instance using Enterprise Manager:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.
2. From the **Availability** menu, select **Cluster Database Operations**, and then select **Delete Instance**.

   The Edit Server Pool page appears.
3. Enter the credentials for the Oracle Database software owner and the SYSASM user, then click **Next**.

   The Delete Instance: Database Instance page appears
4. Modify the server pool settings then click **Submit**.

5. From the **Cluster Database** menu, select **Home** to return to the Cluster Database Home page.
6. Review the number of instances available for your cluster database.

> **See Also:**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

11.7.2 Deleting an Instance From an Administrator-Managed Database

To remove an instance from an administrator-managed database, you must specify the name of the database instance to delete and which node the instance is running on.

To delete an administrator-managed database instance using Enterprise Manager:

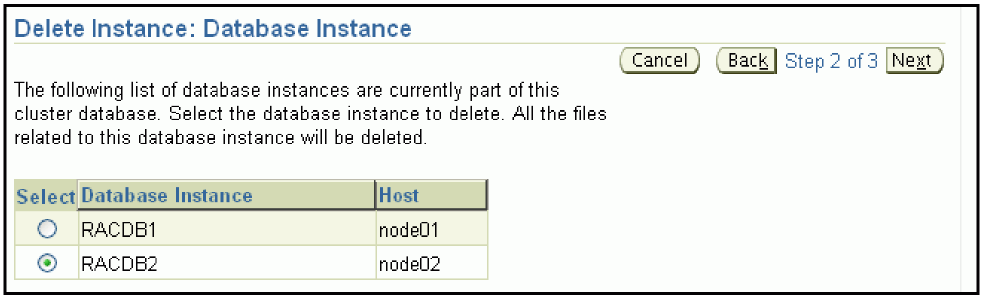
1. In Oracle Enterprise Manager, go to the Cluster Database Home page.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.
2. From the **Availability** menu, select **Cluster Database Operations**, and then select **Delete Instance**.

   The Delete Instance: Cluster Credentials page appears.
3. Enter the credentials for the Oracle Database software owner and the SYSASM user, then click **Next**.

   The Delete Instance: Database Instance page appears
4. Select the instance you want to delete, then click **Next**.

Figure 11-2 Delete Database Instance Page of Oracle Enterprise Manager



Delete Instance: Database Instance

The following list of database instances are currently part of this cluster database. Select the database instance to delete. All the files related to this database instance will be deleted.

Select	Database Instance	Host
○	RACDB1	node01
●	RACDB2	node02

After the host information has been validated, the Delete Instance: Review page appears.

5. Review the information, and if correct, click **Submit Job** to continue. Otherwise, click **Back** and correct the information.

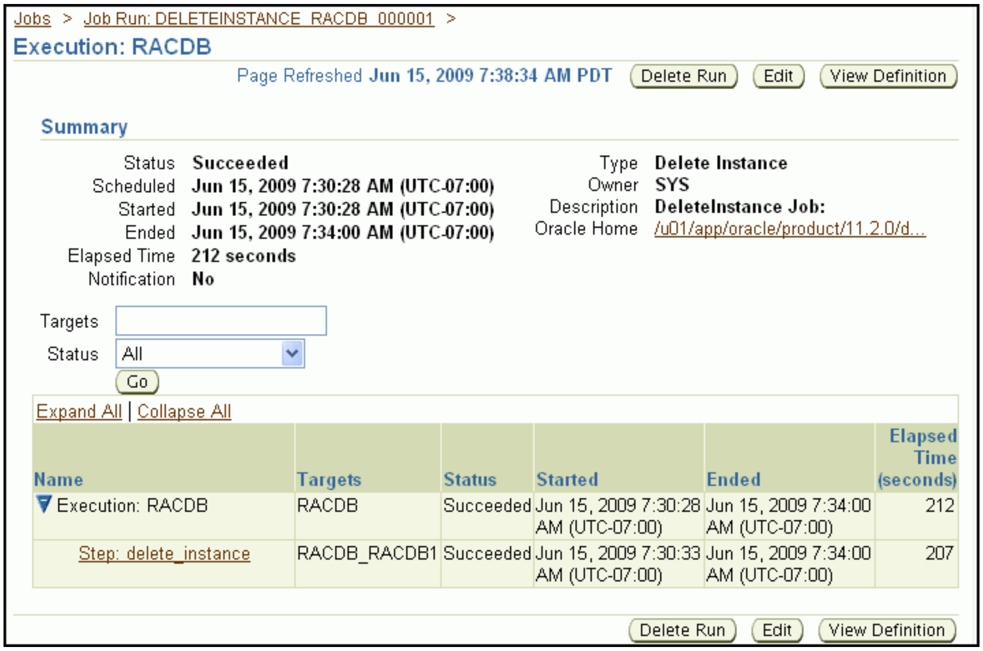
   A Confirmation page appears.

6. Click **View Job** to view the status of the node deletion job.

   A Job Run detail page appears.

7. Click your browser's Refresh button until the job shows a status of Succeeded or Failed.

Figure 11-3 Job Run Detail Page of Oracle Enterprise Manager


Jobs > Job Run: DELETEINSTANCE_RACDB_000001 >
Delete Run
Edit
View Definition
Go
Expand All | Collapse All
Delete Run
Edit
View Definition

Execution: RACDB

Page Refreshed **Jun 15, 2009 7:38:34 AM PDT**

Summary

Status	**Succeeded**	Type	**Delete Instance**
Scheduled	**Jun 15, 2009 7:30:28 AM (UTC-07:00)**	Owner	**SYS**
Started	**Jun 15, 2009 7:30:28 AM (UTC-07:00)**	Description	**DeleteInstance Job:**
Ended	**Jun 15, 2009 7:34:00 AM (UTC-07:00)**	Oracle Home	/u01/app/oracle/product/11.2.0/d...
Elapsed Time	**212 seconds**		
Notification	**No**		

Targets

Status All

Name	Targets	Status	Started	Ended	Elapsed Time (seconds)
▼ Execution: RACDB	RACDB	Succeeded	Jun 15, 2009 7:30:28 AM (UTC-07:00)	Jun 15, 2009 7:34:00 AM (UTC-07:00)	212
Step: delete_instance	RACDB_RACDB1	Succeeded	Jun 15, 2009 7:30:33 AM (UTC-07:00)	Jun 15, 2009 7:34:00 AM (UTC-07:00)	207

If the job shows a status of Failed, then you can click the name of the step that failed to view the reason for the failure.

8. From the **Cluster Database** menu, select **Home** to return to the Cluster Database Home page.

   The number of available instances in the cluster database is reduced by one.

> **See Also:**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

11.8 Removing a Node From the Cluster

Removing a node from the cluster can be as easy as simply shutting down the server.

If the node was not pinned and does not host any Oracle databases using Oracle Database 11*g* release 1 or earlier, then the node is automatically removed from the cluster when it is shut down. If the node was pinned or if it hosts a database instance from earlier releases, then explicit deletion is needed.

To repurpose this server, you can restart the node with a different profile in place, or you can use other software to put a new software image on the server.

See Also:

Oracle Real Application Clusters Administration and Deployment Guide for more information about adding and removing nodes from your cluster database

12
Managing Oracle Software and Applying Patches

To manage Oracle software and apply patches in Oracle Real Application Clusters (Oracle RAC) environments, you use Oracle Universal Installer or the OPatch utility.

- About Patches and Patch Sets
  Oracle issues product fixes for its software called **patches**. They are associated with particular releases and versions of Oracle products.
- Alternative Methods of Patching
  OPatch is a commonly used method for patching Oracle software homes, but this is not the only method of patching Oracle software.
- Obtaining the Patch
  You obtain patches and patch sets from My Oracle Support, which is the Oracle Support Services website.
- Preparing to Use OPatch
  Before you apply the patch to your Oracle RAC database, your Oracle ASM installation, or to your Oracle Clusterware installation, you must prepare the operating system environment and stage the patch locally.
- Applying Patches
  Patching in an Oracle RAC environment is slightly different compared to patching a single node. If OPatch detects a cluster, then it uses OUI to query the software inventory to find the local node name and node list.
- Applying Patch Sets
  For instructions on applying the latest patch set to your Oracle RAC database and Oracle Clusterware installations on Oracle Linux, search for "Oracle 12*c* Release 2 (12.2) Support Status and Alerts" documentation on the My Oracle Support website.
- Troubleshooting Patch Deployment
  Applying patches to your Oracle RAC environment can be complicated at times and you may be required to troubleshoot the patch deployment.
- Upgrading the Oracle Software
  Upgrades are performed when you install a new software release or version, or apply a patch set.

> **See Also:**
>
> - *Oracle Database 2 Day DBA* for more information about software management and patch releases
> - *Oracle OPatch User's Guide for Windows and UNIX* for details on how to get the latest version of OPatch

12.1 About Patches and Patch Sets

Oracle issues product fixes for its software called **patches**. They are associated with particular releases and versions of Oracle products.

The patching cycle involves downloading patches, applying patches, and verifying the applied patch to ensure that the bug fixes present in the patch reflect appropriately.

Patching involves migrating from one version of the software product to another, within a particular release, unlike upgrading which involves moving from one release of a product to another newer release of the software product. When you apply the patch to your 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 periodically releases the following types of patches to fix the bugs encountered in the Oracle software:

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 re-certification.
Security Patch Updates	A cumulative collection of security bug fixes. Previously known as Critical Patch Updates, or 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 and also 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.

12.2 Alternative Methods of Patching

OPatch is a commonly used method for patching Oracle software homes, but this is not the only method of patching Oracle software.

These alternative methods are only described at a conceptual level in this guide. However, you might find these methods more appropriate for your organization than using OPatch.

- Using Oracle Enterprise Manager Cloud Control for Patching Operations
  Using Cloud Control with its Provisioning & Patching functionality, you can automate the patching of your Oracle Grid Infrastructure and Oracle RAC software.
- Rapid Home Provisioning, Patching, and Upgrading
  Rapid Home Provisioning is a method of deploying software homes to any number of nodes in a data center from a single cluster, and also facilitates patching and upgrading software.

12.2.1 Using Oracle Enterprise Manager Cloud Control for Patching Operations

Using Cloud Control with its Provisioning & Patching functionality, you can automate the patching of your Oracle Grid Infrastructure and Oracle RAC software.

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 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

Details on how to perform these tasks, and how to patch your Oracle Grid Infrastructure and Oracle RAC software using Cloud Control are available from the following PDF file: http://www.oracle.com/technetwork/oem/pdf/512066.pdf

The rest of this chapter describes how to install patches without using Cloud Control.

12.2.2 Rapid Home Provisioning, Patching, and Upgrading

Rapid Home Provisioning is a method of deploying software homes to any number of nodes in a data center from a single cluster, and also facilitates patching and upgrading software.

With Rapid Home Provisioning, you create, store, and manage templates of Oracle homes as **images** (called gold images) of Oracle software, such as databases, Oracle Grid Infrastructure, middleware, and applications. You can make a working copy of any gold image, and then you can provision that working copy to any node in the data center.

You store the gold images in a repository located on a **Rapid Home Provisioning Server**, which runs on one server in the Rapid Home Provisioning Server cluster that is a highly available provisioning system. This provisioning method simplifies quick patching of Oracle homes, thereby minimizing or completely eliminating downtime.

Using Rapid Home Provisioning, you can provision Oracle Database software for releases Oracle Database 11*g* release 2 (11.2.0.4) and later. You can also use Rapid Home Provisioning to provision Oracle Grid Infrastructure (11*g* release 2 (11.2.0.4), and 12*c* release 1 (12.1) and release 2 (12.2)) homes and can also be used to patch and upgrade Oracle Grid Infrastructure homes. In addition, there is a rollback capability that facilitates reverting from a patching procedure.

Starting with Oracle Database 12*c* Release 2 (12.2), RHP can manage Oracle Grid Infrastructure and Oracle Database 11.2 and 12.1 software installations. With a single command, RHP can provision new homes to such targets. RHP can also orchestrate the patch or upgrade process for these targets, also with a single command. There is no prerequisite for the targets; you do not need to install any client or agent software on the targets before the provision, patch, or upgrade takes place.

When you provision Oracle Database software, Rapid Home Provisioning offers additional features for creating various types of databases (such as Oracle Real Application Clusters (Oracle RAC), single instance, and Oracle Real Application Clusters One Node (Oracle RAC One Node) databases) on different types of storage, and other options, such as using templates and creating container databases (CDBs). Rapid Home Provisioning improves and makes more efficient patching of database software, allowing for rapid and remote patching of the software, in most cases, without any downtime for the database.

Rapid Home Provisioning simplifies maintenance in large environments because you have only to update software homes one time on the Rapid Home Provisioning Server. Rapid Home Provisioning enables you to create a gold image from any software home. You can then provision that software to any Rapid Home Provisioning Client or target as a working copy. The software may be any binary that you will run on a Rapid Home Provisioning Client or target.

See Also:

Oracle Clusterware Administration and Deployment Guide

12.3 Obtaining the Patch

You obtain patches and patch sets from My Oracle Support, which is the Oracle Support Services website.

The Oracle Support Services website is located at: https://support.oracle.com

To locate patches on the My Oracle Support website:

1. Log in to your account on My Oracle Support.
2. Select the **Patches & Updates** tab.
3. If you know the patch number, then you can enter it into the Patch Name or Number field, then click **Search**.

If you want to search for all available patches for your system, then select **Product or Family (Advanced Search)**, which is located above the Patch Name or Number field. Supply the following information:

- Choose the products you want to patch (for example, Oracle Clusterware, Oracle Database, or an individual product such as Universal Installer)
- Specify the software release for the products you selected, for example, Oracle 12.2.0.1.0.
- Specify the platform on which the software is installed.

Click **Search** to look for available patches.

The Patch Search Results page is displayed.

4. On the Patch Search Results page, select the number of the patch you want to download.

   A details page for that patch appears on your screen.

5. Click the **ReadMe** button to view the ReadMe file for the patch, or click **Download** to download the patch to your local computer.

   If you download the patch to a computer that is not a node in the cluster, then you must transfer the file using a binary protocol to a cluster node.

   You can also choose to add the patch to your current patch plan or create a new patch plan. A *patch plan* is a collection of patches that you want to apply as a group. To learn more about using patch plans, in the Add to Plan drop-down list select **Why use a Plan?**

> **See Also:**
>
> - "Verifying My Oracle Support Credentials"
> - "About Downloading and Installing Oracle Database Patches"
> - *Oracle Database 2 Day DBA* for more information about software management and patch releases

12.4 Preparing to Use OPatch

Before you apply the patch to your Oracle RAC database, your Oracle ASM installation, or to your Oracle Clusterware installation, you must prepare the operating system environment and stage the patch locally.

- Checking the ORACLE_HOME Environment Variable
  OPatch verifies if the Oracle home is present.
- Performing a Backup Before Patching Software
  Oracle strongly recommends that you back up the software directory you are patching before performing any patch operation. This applies to Oracle Database or Oracle Grid Infrastructure software installation directories.

- Staging the Patch on Each Node
  After you download the patch from My Oracle Support, you must copy the patch to each node that runs the software being patched.
- Updating the PATH Environment Variable
  You can either specify the path to the opatch executable file when running OPatch, or you can update the `PATH` environment variable to include the `OPatch` directory.
- Configuring SSH User Equivalency
  User equivalency exists when the operating system user on each node is considered to be identical.

> **See Also:**
>
> - *Oracle Database 2 Day DBA* for more information about how to keep your Oracle Database software up-to-date
> - *Oracle OPatch User's Guide for Windows and UNIX*

12.4.1 Checking the ORACLE_HOME Environment Variable

OPatch verifies if the Oracle home is present.

You must ensure that the `ORACLE_HOME` environment variable is set to the Oracle home of the product you are trying to patch. Also, you must use a version of OPatch that has the same version number as the Oracle home being patched.

Check the respective vendor documentation for the details to set the environment variable.

To check the current setting of the ORACLE_HOME variable on Linux:

1. In a command window, log in to the operating system as the `grid` or `oracle` user.
2. Use the `echo` command to display the current setting of the `ORACLE_HOME` environment variable.

   ```
   echo $ORACLE_HOME
   ```

> **See Also:**
>
> "Configuring the Operating System Environment"

12.4.2 Performing a Backup Before Patching Software

Oracle strongly recommends that you back up the software directory you are patching before performing any patch operation. This applies to Oracle Database or Oracle Grid Infrastructure software installation directories.

To back up the software installation:

1. Back up the software installed in the specified Oracle home using any of the following:
   a. An operating system utility, such as `zip`, `cp -r`, `tar`, or `cpio`, to back up the software in the Oracle home directory to disk.
   b. The Oracle Secure Backup utility to back up the software in the Oracle home directory to tape.

12.4.3 Staging the Patch on Each Node

After you download the patch from My Oracle Support, you must copy the patch to each node that runs the software being patched.

12.4.4 Updating the PATH Environment Variable

You can either specify the path to the opatch executable file when running OPatch, or you can update the `PATH` environment variable to include the `OPatch` directory.

The `opatch` binary file is located in the *`Oracle_home`*`/OPatch` directory.

To update the PATH environment variable on Oracle Linux systems:

1. In a command window, log in to the operating system as the Oracle home owner, or a user that has write permission for the files in the Oracle home.
2. Use a shell command similar to the following to update the value of the PATH environment variable, where `/u01/app/oracle/product/12.2.0/dbhome_1` is the location of your Oracle home directory:

   ```
   $ export PATH=$PATH:/u01/app/oracle/product/12.2.0/dbhome_1/OPatch
   ```

   You could also modify the shell profile script for the current user to have this variable configured every time you log in.

> **See Also:**
>
> "Configuring the Operating System Environment"

12.4.5 Configuring SSH User Equivalency

User equivalency exists when the operating system user on each node is considered to be identical.

For information on configuring user equivalency, see "Configuring Operating System Users and Groups on Linux Systems".

- Testing SSH User Equivalency
  Use the operating system SSH command to verify the user equivalency is working.
- Enabling SSH User Equivalency
  Before you patch a system, make sure the user equivalency is working.

12.4.5.1 Testing SSH User Equivalency

Use the operating system SSH command to verify the user equivalency is working.

1. On the system where you want to run OPatch, log in as the software owner. If you are patching Oracle Grid Infrastructure, login as the `grid` user. If you are patching Oracle RAC software, then login as the `oracle` user.
2. Use the following command to test user equivalency:

   ```
   [oracle@racnode1] $ ssh racnode2 date
   ```

   a. If the date is returned, then user equivalency between the source and destination node has been configured.

   b. If you see output similar to the following, then SSH user equivalency is not enabled:

   ```
   Enter passphrase for key '/home/oracle/.ssh/id_rsa':
   ```

3. If you determine from this test that SSH user equivalency does not exist, then enable SSH user equivalency before continuing with the patching operation.

> **See Also:**
>
> - "Configuring Operating System Users and Groups on Linux Systems"
> - *Oracle Grid Infrastructure Installation and Upgrade Guide* for your specific operating system for instructions on how to configure SSH

12.4.5.2 Enabling SSH User Equivalency

Before you patch a system, make sure the user equivalency is working.

To enable SSH user equivalency:

1. On the system where you want to run OPatch, open a command window and log in as the software owner. Login as `grid` if patching Oracle Grid Infrastructure software, or login as `oracle` if patching Oracle Database software.
2. Start the SSH agent and load the SSH keys into memory using the following commands:

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

These commands start the `ssh-agent` on the local node, and load the RSA and DSA keys into the current session's memory so that you are not prompted to use pass phrases when issuing SSH commands.

3. At the prompt, enter the pass phrase for each key that you generated when configuring Secure Shell, for example:

```
[oracle@racnode1 .ssh]$ exec /usr/bin/ssh-agent $SHELL
[oracle@racnode1 .ssh]$ /usr/bin/ssh-add
Enter passphrase for /home/oracle/.ssh/id_rsa  passphrase
Identity added: /home/oracle/.ssh/id_rsa (/home/oracle/.ssh/id_rsa)
Identity added: /home/oracle/.ssh/id_dsa (/home/oracle/.ssh/id_dsa)
```

4. To test if you have configured SSH correctly, run the following command.

```
[oracle@racnode1] $ ssh racnode2 date
```

If you have configured SSH correctly, then you are not prompted for a password or a pass phrase.

> **Important:**
>
> Do not close this command window until you have completed the patch installation. If you must close the command window in which you enabled SSH user equivalency before the patch installation is complete, then repeat Step 1 to Step 4 before starting the patch installation.

> **See Also:**
>
> - "Configuring Operating System Users and Groups on Linux Systems"
> - *Oracle Grid Infrastructure Installation and Upgrade Guide* for your specific operating system for instructions on how to configure SSH

12.5 Applying Patches

Patching in an Oracle RAC environment is slightly different compared to patching a single node. If OPatch detects a cluster, then it uses OUI to query the software inventory to find the local node name and node list.

Before you install a patch, you must stop all the applications running from the software directory that is being patched. In a cluster, you may have to shut down additional applications, depending upon which software is being patched. The following table lists the applications to stop when patching Oracle software.

Table 12-1 Patching Oracle Home Directories

Oracle Home Directory	Applications to Stop
Oracle RAC Database	Oracle RAC database, Enterprise Manager Agent, listener, and any other applications that run from the Oracle RAC home directory

Table 12-1 (Cont.) Patching Oracle Home Directories

Oracle Home Directory	Applications to Stop
Oracle Grid Infrastructure	Oracle RAC database and all applications running from the Oracle RAC home directory, any single-instance databases that use the same Oracle ASM instance as the cluster database, the Oracle ASM instance, all node applications, Oracle Clusterware, and any other applications that run from the Grid home directory

- Patching the Grid Home
  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.
- All Node Patching
  In all node patching, all the nodes in the cluster are initially shut down and the patch is applied on all the nodes.
- Out-of-Place Rolling Patching for Oracle RAC
  Out-of-place rolling patching involves installing the patch in a new home, modifying the Oracle home of the database, and then restarting instances in a rolling fashion.
- Minimum Downtime Patching
  Minimum downtime patching shortens the time that all the nodes have to be down to apply a patch.

> **See Also:**
>
> - "Obtaining the Patch"
> - "Preparing to Use OPatch"
> - *Oracle Universal Installer Installation Guide*
> - *Oracle OPatch User's Guide for Windows and UNIX* for information about patch automation

12.5.1 Patching the Grid Home

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 patch 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. As the `root` user, perform the prepatch steps for the Grid home using commands similar to the following:

   ```
   cd /u01/app/12.2.0/grid/crs/install
   rootcrs.sh -prepatch
   ```

   When you run the `rootcrs` script with the `-prepatch` option, 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, using one of the patching methods.
3. After you have finished modifying the Grid home, perform the postpatch steps as the `root` user using commands similar to the following:

   ```
   cd /u01/app/12.2.0/grid/crs/install
   rootcrs.sh -postpatch
   ```

   The `rootcrs` script with the `-postpatch` option locks the Grid home again and restarts the Oracle Clusterware stack.

12.5.2 All Node Patching

In all node patching, all the nodes in the cluster are initially shut down and the patch is applied on all the nodes.

After all the nodes have been patched, then the Oracle Clusterware stack and all its registered resources are restarted on each node. This method is typically used for very critical patches and it leads to maximum downtime. OPatch uses this method if the patch cannot be applied in a rolling fashion, and you did not specify the `minimize_downtime` option.

See Also:

- "Obtaining the Patch"
- "Preparing to Use OPatch"
- "Applying Patch Sets"

12.5.3 Out-of-Place Rolling Patching for Oracle RAC

Out-of-place rolling patching involves installing the patch in a new home, modifying the Oracle home of the database, and then restarting instances in a rolling fashion.

Rolling patching is performed group by group, separately, until all the nodes in the cluster are patched. This is the most efficient means of applying an interim patch to an Oracle RAC or Oracle Grid Infrastructure for a cluster installation. By patching groups of nodes individually, there is zero downtime for the cluster database because at least one instance is always available on a different node.

While most patches can be applied in a rolling fashion, some patches cannot be applied in this fashion. The `README` file for the patch indicates whether you can apply the patch using the rolling patch method. If the patch cannot be applied using the rolling patch method, then you must use either "Minimum Downtime Patching" or "All Node Patching" to apply the patch.

When you perform out-of-place patching, you install the updated or patched software into a new Oracle home. Then you update Oracle Database to use the new Oracle home.

1. Install the patched software in a new Oracle home directory on one node.

During installation, the installer detects that this installation is for a cluster and copies the new Oracle home to each node in the cluster.

2. Modify the Oracle home location for the Oracle RAC database using SRVCTL.

   ```
   srvctl modify database -db database_name -oraclehome patched_home_location
   ```

3. Stop all Oracle Database processes on this node.

   ```
   srvctl stop home -oraclehome patched_home_location -node nodename
   ```

4. Start all Oracle Database processes on this node.

   ```
   srvctl start home -oraclehome patched_home_location -node nodename
   ```

   You must restart the Oracle Database processes for them to start using the patched software.

5. Repeat Step 3 and Step 4 on each node in the cluster to restart all the Oracle RAC database instances.

> **See Also:**
>
> - "Obtaining the Patch"
> - "Preparing to Use OPatch"

12.5.4 Minimum Downtime Patching

Minimum downtime patching shortens the time that all the nodes have to be down to apply a patch.

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

When you use the minimum downtime 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 the patched nodes (the first set of nodes) before patching the remaining nodes.

See Also:

- "Obtaining the Patch"
- "Preparing to Use OPatch"
- "Applying Patch Sets"

12.6 Applying Patch Sets

For instructions on applying the latest patch set to your Oracle RAC database and Oracle Clusterware installations on Oracle Linux, search for "Oracle 12*c* Release 2 (12.2) Support Status and Alerts" documentation on the My Oracle Support website.

This document provides a summary of the patch sets available for Oracle 12*c* Release 2. Using this document, you can easily locate and view the Patch Set Notes for your platform. The Oracle Database Patch Set Notes document contains the following information:

- System requirements and information about how to install or reinstall the patch set
- A list of all bugs fixed to date that are specific to Oracle Database for specified platform
- A list of known issues relating to Oracle Database for the specified platform

To locate the Patch Set notes on My Oracle Support:

1. Log in to My Oracle Support.
2. Select the **Patches & Updates** tab.
3. Select **Quick Links to the Latest Patchsets, Mini Packs, and Maintenance Packs**.
4. Under the heading Latest Oracle Server/Tools Patchsets, select **Oracle Database**.

   A list of operating systems appears.
5. Place your cursor over the entry that matches your operating system, or use the triangular arrows to search for your operating system.

   When you place the cursor over the entry for your operating system, for example, Linux x86, a list of database versions appears.
6. Select **12.2.0**

   The Advanced Search page appears.
7. Scroll to the bottom of this page to see the list of available patch sets.
8. Select the number in the Patch column for the patch set you want to view or download.

   The Patchset description and download page appears.
9. Click **View Readme** to see the patch set notes.

   On this page you can also click **Download** to download the patch to your computer.

10. If you choose to download the patch, then follow the instructions in the ReadMe file of the patch set to apply the patch set to your software.

> **See Also:**
>
> "Preparing to Use OPatch"

12.7 Troubleshooting Patch Deployment

Applying patches to your Oracle RAC environment can be complicated at times and you may be required to troubleshoot the patch deployment.

If you have problems applying a patch to your Oracle RAC database, then review these solutions to common problems. If the problem you encountered is not listed, then review the log and trace files and use My Oracle Support to obtain assistance.

- Updating the Node List for OPatch
  If OPatch does not automatically detect Oracle RAC or its nodes, then investigate the contents of the inventory and ensure they are complete.
- About OPatch Log and Trace Files
  Logging and tracing is a common aid for debugging. OPatch maintains logs for all `apply`, `rollback`, and `lsinventory` operations.
- Resolving the "Not a valid patch area" Error
  You might get this error if the directory that the OPatch utility is using to do the patch does not match the template for what it is checking, or if the OPatch utility is run from an invalid directory.
- Resolving the "Unable to remove a partially installed interim patch" Error
  If the patching process is interrupted, then you might get the error "Unable to remove a partially installed interim patch" when you try to install the patch a second time.

> **See Also:**
>
> - *Oracle OPatch User's Guide for Windows and UNIX*
> - "Preparing to Use OPatch"
> - "Viewing Oracle RAC Database Alert Log Messages"
> - "About the Oracle Clusterware Alert Log"

12.7.1 Updating the Node List for OPatch

If OPatch does not automatically detect Oracle RAC or its nodes, then investigate the contents of the inventory and ensure they are complete.

To update the node list for OPatch:

- If the list of nodes for your cluster is not complete, then you can update it by using Oracle Universal Installer and the `-updateNodeList` flag, as demonstrated in the following example:

  ```
  Oracle_home/oui/bin/runInstaller -updateNodeList
  ORACLE_HOME=/u01/app/oracle/product/12.2.0/dbhome_1
  CLUSTER_NODES=racnode1,racnode2,racnode3 -noClusterEnabled
  ```

> **See Also:**
>
> *Oracle Universal Installer Installation Guide* for more information about updating the node list

12.7.2 About OPatch Log and Trace Files

Logging and tracing is a common aid for debugging. OPatch maintains logs for all `apply`, `rollback`, and `lsinventory` operations.

The log files are located in the *`Oracle_home`*`/cfgtoollogs/opatch` directory. 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 June 9, 2016 at 10:57 PM, then it is named as follows:

```
opatch2016-06-09_22-07-57PM_1.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 : Thu Jun 09 22:07:45 MDT 2016
Oracle Home : /u01/app/oracle/product/12.2.0/dbhome_1
OPatch Ver. : 12.2.0.1.0
Current Dir : /u01/app/oracle/product/12.2.0/dbhome_1
Command     : lsinventory -xml /u01/app/oracle/product/12.2.0/dbhome_1 ...
 Log File    : /u01/app/oracle/product/12.2.0/dbhome_1/cfgtoollogs/opatch/
opatch2016-06-09_22-07-57PM_1.log
```

> **See Also:**
>
> *Oracle OPatch User's Guide for Windows and UNIX*

12.7.3 Resolving the "Not a valid patch area" Error

You might get this error if the directory that the OPatch utility is using to do the patch does not match the template for what it is checking, or if the OPatch utility is run from an invalid directory.

The *`Patch_Shiphome`* directory should have the following structure:

- An `etc` subdirectory that has the metadata files
- A `files` subdirectory that has the patch files
- The `etc/config/inventory.xml` file and the `actions.xml` file under the same directory

To resolve the "Not a valid patch area" error:

- Perform one of the following actions:
  1. Remove the patch shiphome directory and re-create it with the proper structure (by extracting the files again).
  2. Start the OPatch utility from the directory where the patch to be installed has been unzipped and staged on disk.
  3. Use the following command when starting OPatch:

     ```
     opatch apply /Patch_Shiphome
     ```

     where *`Patch_Shiphome`* is the location where the patch has been staged on disk.

> **See Also:**
>
> *Oracle OPatch User's Guide for Windows and UNIX*

12.7.4 Resolving the "Unable to remove a partially installed interim patch" Error

If the patching process is interrupted, then you might get the error "Unable to remove a partially installed interim patch" when you try to install the patch a second time.

To resolve the partially installed patch error:

1. Ensure that the environment variable `ORACLE_HOME` is set to the Oracle home directory you are attempting to patch.
2. Go to the *`Oracle_home`*`/.patch_storage/`*`patch-id_timestamp`* directory and run the `restore.sh` script (or `restore.bat` on Windows platforms) as follows:

```
Oracle_home/.patch_storage/patch-id_timestamp/restore.sh
```

3. On Linux and UNIX systems, use the *Oracle_home*/.patch_storage/*patch-id_timestamp*/make.txt file (if available) to modify your operating system environment, as follows:

```
/bin/sh make.txt
```

4. Attempt to apply the patch again.

> **See Also:**
>
> *Oracle OPatch User's Guide for Windows and UNIX*

12.8 Upgrading the Oracle Software

Upgrades are performed when you install a new software release or version, or apply a patch set.

An **upgrade** modifies most, if not all, of the installed Oracle software files, whereas a patch typically modifies only a few files. You can perform rolling upgrades, or you can shut down the Oracle Database and Oracle Clusterware software to perform the upgrade.

You perform out-of-place upgrades when upgrading the software to a new release. To perform an out-of-place upgrade, you install the Oracle Grid Infrastructure for a cluster into a new Grid home. When performing the upgrade, you specify the location of the new Grid home instead of selecting the existing software location.

When performing an out-of-place upgrade, the old and new version of the software are present on the nodes at the same time, each in a different home location, but only one version of the software is active at any given time. To upgrade your Oracle Database 11*g* software or Oracle Database 12*c* release 1 (12.1) software to Oracle Database 12*c* release 2 (12.2), you install Oracle Grid Infrastructure for a cluster and Oracle Database into new Oracle home directories. At the end of the upgrade process, the active version of the software is changed to use the Oracle Grid Infrastructure for a cluster or Oracle Database 12*c* release 2 (12.2) software.

You can use Database Upgrade Assistant (DBUA) to upgrade an existing database to the current release of Oracle Database. Database Upgrade Assistant (DBUA) guides you through the upgrade process and configures your database for the new release. DBUA automates the upgrade process and makes appropriate recommendations for configuration options such as tablespaces and online redo log files.

See Also:

- *Oracle Database 2 Day DBA* for more information about using DBUA to upgrade a database
- *Oracle Grid Infrastructure Installation and Upgrade Guide* for your operating system for more information on performing software upgrades
- *Oracle Database Upgrade Guide*

Glossary

administrator-managed database

An administrator-managed database is a database created on nodes that are not part of a server pool; the database or clusterware administrator manages these databases.

affinity

Any strategy that is expected to increase the probability that an application or client finds the requested data cached in the instance to which the request is routed.

affinity hint

A flag that indicates if affinity is active or inactive for a particular instance and service combination.

all node patching

A method of applying patches to the nodes in a cluster. When using the all node patching method, all the nodes that are part of the Oracle Real Application Clusters database are initially brought down and the patch is applied on all the nodes. After the patch is applied to all nodes, then the nodes are brought back up.

Application Continuity

A feature that can reestablish database sessions and resubmit pending work for recoverable errors, thereby protecting applications from transient failures. When a component fails, the recovery restores the state as it was before the failure. A user request is executed on time and at most once.

Automatic Workload Repository (AWR)

A built-in repository that exists in every Oracle Database. At regular intervals, Oracle Database makes a snapshot of all of its vital statistics and workload information and stores them in the AWR.

cache coherency

The synchronization of data in multiple caches so that reading a memory location through any cache returns the most recent data written to that location through any other cache. Sometimes called cache consistency.

Cache Fusion

A diskless cache coherency mechanism in Oracle Real Application Clusters that provides copies of blocks directly from a holding instance's memory cache to a requesting instance's memory cache.

cluster

Multiple interconnected computers or servers that appear as if they are one server to end users and applications.

cluster database

The generic term for an Oracle Real Application Clusters database.

cluster file system

A distributed file system that is a cluster of servers that collaborate to provide high performance service to their clients. Cluster file system software deals with distributing requests to storage cluster components.

Cluster Synchronization Services (CSS)

An Oracle Clusterware component that discovers and tracks the membership state of each node by providing a common view of membership across the cluster. CSS also monitors process health, specifically the health of the database instance. See also OCSSD.

Cluster Verification Utility (CVU)

A tool that verifies a wide range of Oracle RAC-specific components such as shared storage devices, networking configurations, system requirements, Oracle Clusterware, groups, and users.

connect descriptor

A specially formatted description of the destination for a network connection. A connect descriptor contains destination service and network route information. The network route provides, at a minimum, the location of the listener through use of a network address.

CRSD

A Linux or UNIX process that performs high availability recovery and management operations such as maintaining the OCR. Also manages application resources and runs as `root` user (or by a user in the `admin` group on Mac operating system X-based systems) and restarts automatically upon failure.

database services

Entities that you can define in Oracle RAC databases that enable you to group database workloads and route work to the optimal instances that are assigned to offer the service.

Dynamic Host Configuration Protocol (DHCP)

A network application protocol used by devices (DHCP clients) to obtain configuration information for operation in an Internet Protocol network. This protocol reduces system administration workload, enabling devices to be added to the network with little or no manual intervention.

Event Manager (EVM)

The background process that publishes Oracle Clusterware events. EVM scans the designated callout directory and runs all scripts in that directory when an event occurs.

Event Manager Daemon (EVMD)

A Linux or UNIX event manager daemon that starts the `racgevt` process to manage callouts.

Fast Application Notification (FAN)

Applications use FAN to enable rapid failure detection, balancing of connection pools after failures, and re-balancing of connection pools when failed components are repaired. The FAN notification process uses system events that Oracle publishes when cluster servers become unreachable or if network interfaces fail.

Fast Connection Failover

Fast Connection Failover provides high availability to FAN integrated clients, such as clients that use JDBC, OCI, or ODP.NET. If you configure the client to use fast connection failover, then the client automatically subscribes to FAN events and can react to database `UP` and `DOWN` events. In response, Oracle gives the client a connection to an active instance that provides the requested database service.

forced disk write

In Oracle Real Application Clusters, a particular data block can only be modified by one instance at a time. If one instance modifies a data block that another instance needs, then whether a forced disk write is required depends on the type of request submitted for the block.

Free pool

A default server pool used in policy-based cluster and capacity management of Oracle Clusterware resources. The free pool contains servers that are not assigned to any server pool.

General Parallel File System (GPFS)

General Parallel File System (GPFS) is a shared-disk IBM file system product that provides data access from all of the nodes in a homogenous or heterogeneous cluster.

Global Cache Service (GCS)

A process that implements Cache Fusion. It maintains the block mode for blocks in the global role. It is responsible for block transfers between instances. The Global Cache Service employs various background processes such as the Global Cache Service Processes (LMSn) and Global Enqueue Service Daemon (LMD).

Global Cache Service Processes (LMSn)

Processes that manage remote messages. Oracle RAC provides for up to 10 Global Cache Service Processes.

Global Cache Service (GCS) resources

Global resources that coordinate access to data blocks in the buffer caches of multiple Oracle RAC instances to provide cache coherency.

global database name

The full name of the database that uniquely identifies it from any other database. The global database name is of the form *database_name*.*database_domain*—for example: TEST.US.EXAMPLE.COM

global dynamic performance views

Dynamic performance views storing information about all open instances in an Oracle Real Application Clusters cluster. (Not only the local instance.) In contrast, standard dynamic performance views (V$) only store information about the local instance.

Global Enqueue Service (GES)

A service that coordinates enqueues that are shared globally.

Global Enqueue Service Daemon (LMD)

The resource agent process that manages requests for resources to control access to blocks. The LMD process also handles deadlock detection and remote resource requests. Remote resource requests are requests originating from another instance.

Global Enqueue Service Monitor (LMON)

The background LMON process monitors the entire cluster to manage global resources. LMON manages instance deaths and the associated recovery for any failed instance. In particular, LMON handles the part of recovery associated with global resources. LMON-provided services are also known as Cluster Group Services.

Global Services Daemon (GSD)

A component that receives requests from SRVCTL to execute administrative job tasks, such as startup or shutdown. The command is executed locally on each node, and the results are returned to SRVCTL. GSD is installed on the nodes by default.

Grid home

The Oracle Home directory for Oracle Grid Infrastructure for a cluster, which includes Oracle Clusterware and Oracle ASM.

grid infrastructure

The software that provides the infrastructure for an enterprise grid architecture. Oracle Grid Infrastructure for a cluster bundles these infrastructure software products into one product. In an Oracle cluster, Oracle Grid Infrastructure for a cluster includes Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM). For a standalone Oracle Database server, Oracle Grid Infrastructure for a cluster includes Oracle Restart and Oracle ASM.

Grid Naming Service (GNS)

A generic service which resolves the names of hosts in a delegated normal DNS zone by mapping them to IP addresses within the zone. GNS enables the use of Dynamic Host Configuration Protocol (DHCP) address for Oracle RAC database nodes, simplifying deployment. GNS also resolves host names passed back from a SCAN listener.

high availability

Systems with redundant components that provide consistent and uninterrupted service, even following hardware or software failures. This involves some degree of redundancy.

High Availability Cluster Multi-Processing (HACMP)

High Availability Cluster Multi-Processing is an IBM AIX-based high availability cluster software product. HACMP has two major components: high availability (HA) and cluster multi-processing (CMP).

instance

For an Oracle RAC database, each node in a cluster usually has one instance of the running Oracle software that references the database. When a database is started, Oracle allocates a memory area called the System Global Area (SGA) and starts one or more Oracle processes. This combination of the SGA and the Oracle processes is called an instance. Each instance has unique Oracle System Identifier (SID), instance name, rollback segments, and thread ID.

instance membership recovery

The method used by Oracle RAC guaranteeing that all cluster members are functional or active. IMR polls and arbitrates the membership. Any members that do not show a heartbeat by way of the control file or who do not respond to periodic activity inquiry messages are presumed terminated.

instance name

Represents the name of the instance and uniquely identifies a specific instance when clusters share common services names. The `INSTANCE_NAME` parameter in the instance initialization file, `init`*`sid`*`.ora`, identifies the instance name. The instance name equals the Oracle System Identifier (SID).

instance number

A number that associates extents of data blocks with particular instances. The instance number enables you to start an instance and ensure that it uses the extents allocated to it for inserts and updates. This ensures that an instance does not use space allocated for other instances.

interconnect

The private network communication link used to synchronize the memory cache of the nodes in the cluster.

listener

A process that resides on the server and whose responsibility is to listen for incoming client connection requests and manage the traffic to the server. When a client requests a network session with a database server, a listener receives the actual request. If the client information matches the listener information, then the listener grants a connection to the database server.

logical transaction ID

A globally unique identifier that defines a transaction from the application perspective.

Logical Volume Manager (LVM)

A generic term that describes Linux or UNIX subsystems for online disk storage management.

Inter-Process Communication (IPC)

A high-speed operating system-dependent transport component. The IPC transfers messages between instances on different nodes. Also referred to as the interconnect.

Master Boot Record (MBR)

A program that executes when a computer starts. Typically, the MBR resides on the first sector of a local hard disk. The program begins the startup process by examining the partition table to determine which partition to use for starting the computer. The MBR program then transfers control to the boot sector of the startup partition, which continues the startup process.

minimum downtime patching

In minimum downtime patching, the nodes are divided into two sets. The first set is shut down and the patch is applied to it. The second set is then shut down. The first set is brought up and then the patch is applied to the second set. After the patch is applied to the second set, those nodes are also brought up, finishing the patching operation.

multicast Domain Name Server (mDNS)

A part of Zero Configuration Networking (Zeroconf), mDNS provides the ability to address hosts using DNS-like names without the need of an existing, managed DNS server.

mutables

Data returned from mutable objects. An example of a mutable is a call to `SYSTIMESTAMP` function.

mutable objects

A function that obtains a new value every time it is called. Examples of mutable objects are `SYSDATE`, `SYSTIMESTAMP`, `SYS_GUID`, and *`sequence`*`.NEXTVAL`.

net service name

A simple name for a service that resolves to a connect descriptor. Users initiate a connect request by passing a user name and password along with a net service name in a connect string for the database service to which they want to connect.

Network Attached Storage (NAS)

Storage that is attached to a server by way of a network.

Network Interface Card (NIC)

A card that you insert into a computer to connect the computer to a network.

network switch

A hardware device that connects computers within a network.

Network Time Protocol (NTP)

An Internet standard protocol, built on top of TCP/IP, that ensures the accurate synchronization to the millisecond of the computer clock times in a network of computers.

nodes

A node is a computer on which the Oracle Clusterware software is installed or will be installed.

Object Link Manager (OLM)

The Oracle interface that maps symbolic links to logical drives and displays them in the OLM graphical user interface.

OCSSD

A Linux or UNIX process that manages the Cluster Synchronization Services (CSS) daemon. Manages cluster node membership and runs as `oracle` user; failure of this process results in cluster restart.

optimal flexible architecture (OFA)

A set of file naming and configuration guidelines created to ensure reliable Oracle installations that require little maintenance.

Oracle Base directory

The mountpoint for all software installations performed by a particular user. An Oracle base directory can contain multiple Oracle homes for Oracle software products, either of the same or different releases, all installed by the same operating system user. The Oracle Base directory is also the directory where the software parameter files, log files, trace files, and so on, associated with a specific installation owner are located.

Oracle Cluster File System (OCFS)

The Oracle proprietary cluster file system software that is available for Linux platforms.

Oracle Cluster Registry (OCR)

The Oracle RAC configuration information repository that manages information about the cluster node list and instance-to-node mapping information. The OCR also manages information about Oracle Clusterware resource profiles for customized applications.

Oracle Clusterware

This is clusterware that is provided by Oracle to manage cluster database processing including node membership, group services, global resource management, and high availability functions.

Oracle Home directory

The binary location for a particular software installation.

Typically The Oracle Home directory is a subdirectory of the Oracle Base directory for the software installation owner. However, with Oracle Grid Infrastructure for a cluster, the Oracle Home directory (in this case, the Grid home) is located outside of the Oracle Base directory for the Oracle Grid Infrastructure for a cluster installation owner, because the path of the Grid home is changed to `root` ownership.

Oracle Interface Configuration Tool (OIFCFG)

A command-line tool for both single-instance Oracle databases and Oracle RAC databases that enables you to allocate and de-allocate network interfaces to components, direct components to use specific network interfaces, and retrieve component configuration information. The Oracle Universal Installer (OUI) also uses OIFCFG to identify and display available interfaces.

Oracle Inventory directory

The Oracle Inventory directory is the central inventory location for all Oracle software installed on a server.

Oracle Notification Services (ONS)

A publish and subscribe service for communicating information about all FAN events.

Oracle Universal Installer (OUI)

A tool to install Oracle Clusterware, the Oracle relational database software, and the Oracle Real Application Clusters software. You can also use the Oracle Universal Installer to launch the Database Configuration Assistant (DBCA).

policy-managed database

A policy-managed database is created using a server pool. Oracle Clusterware allocates and reassigns capacity based on policies you define, enabling faster resource failover and dynamic capacity assignment.

recoverable error

A class of errors that arise because of an external system failure, independently of the application session logic that is executing. Recoverable errors occur following planned and unplanned outages of networks, nodes, storage, and databases. An example of a nonrecoverable error is submission of invalid data values.

Recovery Manager (RMAN)

An Oracle tool that enables you to back up, copy, restore, and recover data files, control files, and archived redo logs. It is included with the Oracle server and does not require separate installation. You can invoke RMAN as a command line utility from the operating system (O/S) prompt or use the GUI-based Enterprise Manager Backup Manager.

redo thread

The redo generated by a database instance.

request

A sequence of operations on a database connection that uses client and server non-transactional states. A request may include zero or more database transactions. Global transactions can span requests.

request boundary

The beginning or end of a request. A request begins when a connection is borrowed from the Universal Connection Pool (UCP) or WebLogic Server connection pool, and ends when this connection is returned to the connection pool. A request boundary can also be explicitly defined with application calls such as `beginRequest` and `endRequest`. See also request.

rolling patching

In Rolling Patching, one node (or group of nodes) is shutdown, the patch applied and the node brought back up again. This is repeated for each node in the cluster until all the nodes in the Real Application Clusters are patched.

Run-time Connection Load Balancing

Enables Oracle to make intelligent service connection decisions based on the connection pool that provides the optimal service for the requested application based on current workloads. The JDBC, ODP.NET, and OCI clients are integrated with the load balancing advisory; you can use any of these client environments to provide run-time connection load balancing.

scalability

The ability to add additional nodes to Oracle Real Application Clusters applications and achieve markedly improved scale-up and speed-up.

SCAN

A single name, or network alias, for the cluster. Oracle Database 11*g* database clients use SCAN to connect to the database. SCAN can resolve to multiple IP addresses, reflecting multiple listeners in the cluster handling public client connections.

Secure Shell (SSH)

A program for logging into a remote computer over a network. You can use SSH to execute commands on a remote computer and to move files from one computer to another. SSH uses strong authentication and secure communications over insecure channels.

Server Control (SRVCTL) Utility

Server Management (SRVM) comprises the components required to operate Oracle Enterprise Manager in Oracle Real Application Clusters. The SRVM components, such as the Intelligent Agent, Global Services Daemon, and SRVCTL, enable you to manage cluster databases running in heterogeneous environments through an open client/server architecture using Oracle Enterprise Manager.

server pool

A server pool is a logical division of nodes in a cluster into a group to support policy-managed databases.

shared everything

A database architecture in which all instances share access to all of the data.

singleton services

Services that run on only one instance at any one time.

split brain syndrome

Where two or more instances attempt to control a cluster database. In a two-node environment, for example, one instance attempts to manage updates simultaneously while the other instance attempts to manage updates.

system identifier (SID)

The Oracle system identifier (SID) identifies a specific instance of the running Oracle software. For an Oracle Real Application Clusters database, each node within the cluster has an instance referencing the database.

thread

Each Oracle instance has its own set of online redo log groups. These groups are called a thread of online redo. In non-Oracle Real Application Clusters environments, each database has only one thread that belongs to the instance accessing it. In Oracle Real Application Clusters environments, each instance has a separate thread, that is, each instance has its own online redo log. Each thread has its own current log member.

thread number

An identifier for the redo thread to be used by an instance, specified by the `INSTANCE_NUMBER` initialization parameter. You can use any available redo thread number but an instance cannot use the same redo thread number as another instance.

transaction

A logical unit of work that contains one or more SQL statements. All statements in a transaction commit or roll back together. The use of transactions is one of the most important ways that a database management system differs from a file system.

transaction ID

An identifier that is unique to a transaction and represents the undo segment number, slot, and sequence number.

transaction idempotence

The ability to return a guaranteed outcome for a transaction: whether it committed and whether the call was completed.

transparent application failover (TAF)

A run-time failover for high-availability environments, such as Oracle Real Application Clusters, TAF refers to the failover and re-establishment of application-to-service connections. It enables client applications to automatically reconnect to the database if the connection fails, and optionally resume a SELECT statement that was in progress. This reconnect happens automatically from within the Oracle Call Interface (OCI) library.

voting disk

A file that manages information about node membership.

Index