Step by Step installing Oracle 11g RAC on VMware workstation 10/RHL5

Host Physical System: Lenovo T420s, Processor: Intel i5-254M 2.60Hz, RAM: 4 GB
Host OS: Windows 7 Professional, 32-bit

3G switch for XP, Vista and Windows7 32 bits: Normally XP and Vista 32 bit will only address up to 2GB memory for applications and 2GB for the kernel. The 3G swicth will change this to 3GByte for applications and 1GByte for the kernel. This will make VMWare works faster.

Adding large 3G memory to Windows XP, If we add a boot.ini switch /3GB, then XP is allowed to address up to 3GB for applications. Manually edit your boot.ini file by adding " /3GB /Userva=2900" to your boot.ini (no quotes). Your boot.ini is normally read only and you will need to change that temporarily while you edit it. To Edit it manually you can find it on the root of your system drive or go to system properties/advanced tab, settings under 'startup and recovery' section, then hit edit.

Typical : multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Windows XP Professional" /FASTDETECT /NOEXECUTE=OPTIN /3GB /Userva=2900

Yours maybe different, the only thing you should change is add /3GB /Userva=2900.

To enable the 3GB switch on Windows Vista™ or Windows 7:

1. Right-click Command Prompt in the Accessories program group of the Start menu. Click Run as Administrator.
2. At the command prompt, enter **"bcdedit /set IncreaseUserVa 3072"**
3. Restart the computer.

To disable the 3GB switch:

1. Right-click on Command Prompt in the Accessories program group of the Start menu. Click Run as Administrator.
2. At the command prompt, enter **"bcdedit /deletevalue IncreaseUserVa"**
3. Restart the computer.

Note: Its safer, especially if using a video card, framegrabbers with lots of memory (like 640-768MB) to set it to 2.5 or **2.75GB i**nstead of three

VMWare workstation 10 virtual machines are used as oracle cluster nodes. A virtual machine is a software computer that, like a physical machine, runs an operating system and applications. A virtual machine uses the physical resources of the physical machine on which it runs, which is called the host system. Virtual machines have virtual devices that provide the same functionality as physical hardware, but with the additional benefits of portability, manageability and security.

Vitual machine's OS: RedHat Enterprose Linux 5, 64-bit.

Oracle Database 11g Release 2 (11.2) provides the infrastructure for enterprise grid architecture. In a cluster, this software includes Oracle Clusterware and Oracle ASM. For a standalone server, this software includes Oracle Restart and Oracle ASM. This software combines Oracle Clusterware and Oracle ASM into one software installation called the grid infrastructure home.

Understand Oracle Grid Computing

Grid computing: A computing architecture that coordinates large numbers of servers and storage to act as a single large computer, Oracle database service is accessed by a Single Client Access Name (SCAN).

Grid Network Structure

Oracle RAC databases provide the important benefits of connection load balancing and failover. Standalone Oracle Databases perform load balancing by distributing connections among local dispatchers. Oracle RAC databases increase load balancing by using the Oracle Clusterware SCANs (Single Client Access Name) to balance connections among all instances in a cluster database.

Before we introduce Oracle Clusterware SCANs, we need to understand oracle basic network mechanism.

Oracle Network elements

Oracle Net provides the mechanism for launching a `server process` to execute code on behalf of a user process. This is referred to as establishing a `session`. Thereafter, Oracle Net is responsible for maintaining the session: transmitting SQL from the user process to the server process, and fetching results from the server process back to the `user process`.

In release 11g, Oracle's network support is limited to TCP, TCP with secure sockets, Windows Named Pipes (or NMP), and the newer Sockets Direct Protocol (or SDP) over Infiniband highspeed networks. This reduction in protocol support is in line with industry standards. All operating systems also have an Inter-Process Communication (or IPC) protocol proprietary to the operating system—this is also available to Oracle Net for local connections where the user process is on the same machine as the server.

A user logs on to the database by specifying an `Oracle account name` followed by some means of `authentication` (`Operating system authentication`, `Password file authentication`, `Database Password authentication`, `External authentication`, `Global authentication`) establishing a `session` against the `instance` and logging on with the user account name. Associated with a user account is a `schema`, a set of objects owned by a user account in database. The `PMON` background process in `Instance` periodically polls all the `server processes`, to ascertain the state of the session. If a server process reports that it has lost contact with its `user process`, PMON will tidy up. If the session were in the middle of a transaction, PMON will roll back the transaction and release any locks held by the session. Then it will terminate the `server process` and release the `PGA` back to the operating system.

There are three critical files, under *`$ORACLE_HOME/network/admin`*, involved in configuring Oracle Net:

- The **listener.ora** file is a server-side file that defines database listeners. It includes the protocols, addresses, and ports on which they will listen for incoming connection requests, and (optionally) a hard-coded list of instances against which they will launch sessions.
- The **tnsnames.ora** file is a client-side file used for name resolution. It is used by user processes to locate database listeners. It may also be used by the instance itself, to locate a listener with which to register.
- The **sqlnet.ora** file is optional and may exist (possibly with different settings) on the server side, the client side, or both. It contains settings that apply to all connections and listeners, such as security rules and encryption.

Net Manager Tool:

1. Profile: configure sqlnet.ora,
2. Service Name: configure tnsname.ora
3. Listener: configure listener.ora

The database listener, running on the server machine, uses one or more protocols to monitor one or more ports on one or more network interface cards for incoming connection requests, and spawns server processes against an instance.

When a listener receives a connect request from a client, it must first validate whether the instance requested is actually available. Assuming that it is, the listener will launch a new server process to service the user process. Thus if you have a thousand users logging on concurrently to your instance, you will be launching a thousand server processes. This is known as the dedicated server architecture, you'll see the shared server alternative where each user process is handled by a dedicated dispatcher process, but shared by multiple user processes.

In the TCP environment, each dedicated server process launched by a listener will acquire a unique TCP port number. This will be assigned at process startup time by your operating system's port mapping algorithm. The port number gets passed back to the user process by the listener (or on some operating systems the socket already opened to the listener is transferred to the new port number), and the user process can then communicate directly with its server process. The listener has now completed its work and waits for the next connect request.

To create a listener, you need do nothing more than create an entry in the *listener.ora* file, and start it (*lsnrctl*). You can further complicate matters by running multiple listeners on one machine, and any one listener can accept connection requests for a number of instances.

For an instance to register a listener, you can use initialization parameter *local_listener* in spfile, or

```
SQL> alter system set local_listener='(address=(protocol=tcp)(host=127.0.0.1)(port=1521))';
SQL> alter system register;
```

The initialization parameter *local_listener* tells the instance the network address that it should contact to find a listener with which to register. At instance startup time, the PMON process will use this parameter to locate a listener, and inform it of the instance's name and the names of the service(s) that the instance is offering. The instance name is defined by the *instance_name* parameter, and the *service_names* parameter will have defaulted to this suffixed by the *db_domain* parameter, which will default to *null*.

It is possible to create and start additional services at any time, either by changing the value of the *service_names* parameter (which can be a comma-delimited list, if the instance is to offer several services) or programmatically using the *DBMS_SERVICE* package.

You can run a listener (*lsnrctl*) completely on defaults, without a *listener.ora* file at all. It will listen on whatever addresses resolves to the machine's hostname (or ORACLE_HOST environment variable), on port 1521. Always configure the listener.ora file, to make your Oracle Net environment self-documenting.

A listener is necessary to spawn server processes against an instance. In order to do this, it needs to know what instances are available on the computer on which it is running. A listener finds out about instances by the process of "registration". The listener and the instance must be running on the same computer, unless you are using RAC. In a RAC environment, any listener on any computer in the cluster can connect you to any

instance on any computer. There are two methods for registering an instance with a database: static and dynamic registration. For static registration, you hard-code a list of instances in the `listener.ora` file, and Dynamic registration means that the instance itself (release 8i later), at startup time, locates a listener with `local_listener` initial parameter, and registers with it with `instance_name and service_name` parameter.

Client connection: The user process issues connect (or session) by using connection string, that must resolve to the address of a listener and the name of a service (or of an instance).

For example, start sqlplus client program without logon to any listener (instance) firstly, *$sqlplus /nolog*, then

connect user/pwd[@connectString]	Validate the username/password combination against values stored in the data dictionary.
connect user/pwd[@connectString] as sysdba connect user/pwd[@connectString] as sysoper	Instruct Oracle to go to the external password file to validate the username/password combination. *GRANT [sysdba | sysoper] TO username ;* *GRANT [sysdba | sysoper] TO username ;* Granting either (or both) of these privileges will copy the user's password from the data dictionary into the external password file, where it can be read by the instance even if the database is not open. Create the password file by running the orapwd utility from an operating system prompt. This will create the file and embed within it a password for the SYS user.
connect / as sysdba connect / as sysoper	Use operating system authentication as oracle user 'sys'. Normally you do NOT need any password here; the client must be first logged on to the physical server as an operating system user (normally 'oracle') with access to the Oracle binaries (ownership of the Oracle binaries). This is absolutely neccesary to startup instance while there is nowhere to locate username/password in database.

There is a database user name, followed by a database password if you want a database password authentication, and the two are separated by a "/" as a delimiter. Then there is a "@" symbol, followed by a "connectString". The "@" symbol is an indication to the user process that a network connection is required. If the "@" and the connect string are omitted, then the user process will assume that the instance you wish to connect to is running on the local machine, you need to define environment variable in `ORACLE_SID` in SHELL to tell sqlplus which instance to connect to (for windows, `ORACLE_SID` may be defined in register key), and that the always available `IPC` protocol can be used.

Users with role as `SYSOPER` have the ability to issue specific commands listed below:

```
STARTUP
SHUTDOWN
ALTER DATABASE [MOUNT | DISMOUNT| OPEN | CLOSE]
ALTER [DATABASE | TABLESPACE] [BEGIN | END] BACKUP
RECOVER
```

But they can only be enabled when users are connecting with an external authentication method: either operating system or password file. Connection with `operating system` or `password file` authentication is always possible, no matter what state (`idle, nomount, mount, open`) the instance and database are in, and is necessary to issue `STARTUP` or `SHUTDOWN` commands, To use operating system authentication, the OS user, normally named `oracle`, must be first logged on to the physical server where database installed with the ownership and access to the `Oracle binaries`. To enable operating system and password file authentication (the two go together) for an account, you must grant the user either the `SYSDBA` or the `SYSOPER` privilege:

```
GRANT [sysdba | sysoper] TO username;
```

Granting either (or both) of these privileges will copy the user's password from the data dictionary into the external password file, where it can be read by the instance even if the database is not open. The instance authenticates users by checking whether the operating system user attempting the connection is a member of the operating system group that owns the `Oracle_Home` installation, normally named `oracle`, then no passowrd is needed (just `CONNECT / AS [SYSOPER | SYSDBA];`). For operating system users who are NOT a member of the operating system group that owns the `Oracle_Home` installation, the instance will use password file for the authentication. To determine to whom the SYSDBA and SYSOPER privileges have been granted, query the view `v$PWFILE_USERS` (by default, only user `SYS` is included).

Without the Advanced Security Option (a Kerberos server, a RADIUS server), the only form of external authentication that can be used is operating system authentication. This is a requirement for `SYSDBA and SYSOPER` accounts (as already discussed) but can also be used for normal users. The technique is to create an `Oracle user account` with the same name as the `operating system user account` but prefixed with a string specified by the instance parameter `OS_AUTHENT_PREFIX`. This parameter defaults to the string `OPS$`. Under Windows, when Oracle queries the operating system to identify the user, Windows will usually (depending on details of Windows security configuration) return the username prefixed with the Windows domain. Note that the `username must be in uppercase`, and because of the illegal characters (a backslash and a space) must be enclosed in double quotes.

```
C:\>sqlplus / as sysdba

    SQL*Plus: Release 11.2.0.1.0 Production on Tue Dec 30 16:06:42 2014
    Copyright (c) 1982, 2010, Oracle.  All rights reserved.

    Connected to:
    Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
    With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> create user "OPS$LII01\LCHEN" identified externally;
    User created.
SQL> grant create session to "OPS$LII01\LCHEN";
    Grant succeeded.
SQL> select * from v$PWFILE_USERS;
    USERNAME     SYSDB    SYSOP    SYSAS
    --------     -----    -----    -----
    SYS          TRUE     TRUE     FALSE

SQL> grant sysdba to "OPS$LII01\LCHEN";
    grant sysdba to "OPS$LII01\LCHEN"
    *
    ERROR at line 1:
    ORA-01997: GRANT failed: user 'OPS$LII01\LCHEN' is identified externally


C:\>sqlplus /
    SQL*Plus: Release 11.2.0.1.0 Production on Tue Dec 30 16:07:08 2014
    Copyright (c) 1982, 2010, Oracle.  All rights reserved.

    Connected to:
    Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
    With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> show user
    USER is "OPS$LII01\LCHEN"
```

An example of user ‘lchen’, who doesn’t t own the `Orale_Home` binaries, who has to use oracle `password file` to access oracle using sqlplus:

```
[root@linux1 ~]# su - lchen
[lchen@linux1 ~]$ sqlplus
    -bash: sqlplus: command not found, you need to setup oracle environment for this user using ‘oraenv’
[lchen@linux1 ~]$ cd /usr/local/bin
[lchen@linux1 bin]$ ls
    coraenv  dbhome  oraenv
[lchen@linux1 bin]$ . oraenv
    ORACLE_SID = [lchen] ? oralin
    ORACLE_BASE environment variable is not being set since this information is not available for the
    current user ID lchen. You can set ORACLE_BASE manually if it is required.
[lchen@linux1 bin]$ sqlplus / as sysdba
    SQL*Plus: Release 11.2.0.1.0 Production on Wed Jan 14 07:26:34 2015
    Copyright (c) 1982, 2009, Oracle.  All rights reserved.

    ERROR:
    ORA-01031: insufficient privileges


[lchen@linux1 bin]$ sqlplus sys/sys as sysdba
    SQL*Plus: Release 11.2.0.1.0 Production on Wed Jan 14 07:26:45 2015
    Copyright (c) 1982, 2009, Oracle.  All rights reserved.

    Connected to:
    Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
    With the Partitioning, Real Application Clusters, OLAP, Data Mining and Real Application Testing
    options
SQL> show user
    USER is "SYS"
```

All the connect string information is defined in *`$TNS_ADMIN/tnsname.ora`*. Oracle Net has to work out what is meant by “connectString”: the protocol you want to use (assume that this is TCP), the IP address (or hostname) on which the database listener is running, the port that the listener is monitoring for incoming connection requests, and the name of the instance (which need not be the same as the connect string) to which you wish to connect. Rather than specifying an instance by name, the connect string can include the name of **a service**, which (in a RAC environment) could be made up of **a number of instances**. In a single-instance environment, services can still be used, perhaps to assist with tracking the workload imposed on the database by different groups of users. Connect string can also be used to create database link, so you can manipulate data in the database of another instance. like: `create database link prodstore connect to store identified by admin123 using` *`'prod';`* This defines a database link from the current database to a remote database, with username `store`, password `admin123`, identified by the connect string ‘*prod*’. The link exists in and can only be used by the current user’s schema. When a statement such as: `select * from orders@prodstore;` is issued, the user’s session will launch a session against the remote database, log on to it transparently as user `STORE`, and run the query there. The results will be sent back to the local database and then returned to the user.

As a DBA, you can use “easy connect” to connect to a remote listener whose alias name may not be defined on your local naming machnism *`tnsname.ora`*,

`$sqlplus system/oracle@hostname:port/{instancename:servicename}`

User process initiates a connection to database
Is a connect string specified?
False
Is ORACLE_SID specified?
False
Error – Cannot resolve connection string
True
Use ORACLE_SID as connect string
True
Oracle Net Resolution
Is TNS_ADMIN specified?
False
Is ORACLE_HOME specified?
False
Error – Cannot locate Oracle Net client config files
True
True
Use directory specified by TNS_ADMIN to locate sqlnet.ora
Use default location for sqlnet.ora: ORACLE_HOME\network\admin
Consider NAMES.DIRECTORY_PATH list in sqlnet.ora
Is TNSNAMES specified first?
False
Use other names resolution mechanism, like LDAP, EZCONNECT, HOSTNAME, to resolve host:port:sid
True
Locate tnsnames.ora in directory specified by TNS_ADMIN or in the default location, ORACLE_HOME\network\admin, and resolve host:prot:sid
Client – Server connection
Is connection specified with an @
False
A listener will not be used and this connection can only succeed if client is directly connected to server
True
Is LISTENER listening on host:port:sid?
False
Error – Cannot resolve connection string
True
User process tries to connect to instance via LISTENER

Single Client Access Name

The Single Client Access Name (SCAN) is a fully qualified name (hostname+domain) that is configured to resolve to all the addresses allocated for the SCAN. The addresses resolve using Round Robin DNS either on the DNS server, or within the cluster in a GNS configuration. SCAN listeners can run on any node in the cluster.

Oracle Database 11g release 2 and later instances only register with SCAN listeners as remote listeners. Upgraded databases register with SCAN listeners as remote listeners, and also continue to register with all node listeners.

SCANs run on the cluster, not on a cluster node. SCANs provide location independence for the databases, so that client configuration does not have to depend on which nodes are running a particular database. For example, if you configure **policy managed server pools** in a cluster, then the SCAN allows connections to databases in these server pools regardless of which nodes are allocated to the server pool.

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

During Oracle grid infrastructure installation, SCAN listeners are created for as many IP addresses as there are addresses assigned to resolve to the SCAN. Oracle recommends that the SCAN resolves to three IP addresses, to provide high availability and scalability. If the SCAN resolves to three IP addresses, then there are three SCAN listeners created.

Oracle RAC provides failover with the node VIP addresses by configuring multiple listeners on multiple nodes to manage client connection requests for the same database service. If a node fails, then the service connecting to the VIP is relocated transparently to a surviving node, enabling fast notification of the failure to the clients connecting through the VIP. If the application and client are configured with transparent application failover options, then the client is reconnected to the surviving node. Oracle Clusterware routes client application requests from the SCAN to the VIP on the least loaded instance providing the service.

When performing an advanced installation of the Oracle Grid Infrastructure for cluster software, you use Grid Naming Service (GNS) and Dynamic Host Configuration Protocol (DHCP) for virtual IPs (VIPs). Grid Naming Service is a new feature in Oracle Database 11g release 2 that 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).

This guide builds an Oracle Network Cluster Structure without using Grid Naming Service (GNS). You must ask your network guys to configure the following addresses manually in your corporate DNS:

- A public IP address for each node
- A virtual IP address for each node
- Three Single Client Access Name (SCAN) addresses for the cluster

Oracle Clusterware uses interfaces marked as private as the cluster interconnects. 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 (can be on different VLAN) 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.

Interconnect used by Oracle RAC is the same interconnect that Oracle Clusterware uses. Each cluster database instance uses this interconnects for messaging to synchronize the use of shared resources by each instance. Oracle RAC also uses 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 (NIC), or network adapters. One adapter is for the public network interface and the other adapter is for the private network interface (interconnect). Install additional network adapters on a node if that node is using network attached storage (NAS). You should have a separate network adapter for NAS, or you want to use redundant interconnects

  In previous releases, to make use of redundant networks for interconnect; bonding, trunking, teaming, or similar technology was required. Oracle Grid Infrastructure for a cluster and Oracle RAC can now make use of redundant network interconnects, without the use of other network technology, to enhance optimal communication in the cluster. This functionality is available starting with Oracle Database 11*g* Release 2 (11.2.0.2).

  Redundant Interconnect Usage enables load-balancing and high availability across multiple (up to 4) private networks (also known as interconnects).

- Public interface names must be the same for all nodes. If the public interface on one node uses the network adapter eth0, then you must configure eth0 as the public interface on all nodes. Network interface names are case-sensitive.

- You should configure the same private interface names for all nodes as well. If eth1 is the private interface name for the first node, then eth1 should be the private interface name for your second node. 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).

  You must use a switch for interconnect. Oracle recommends that you use a dedicated network switch. Token-rings or crossover cables are not supported for interconnect.

  For the private network, the end points 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.

- Loopback devices are not supported.

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. 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.

Identity	Home Node	Host Node	Given Name	Type	Address	Address Assigned By	Resolved By
Node 1 Public	Node 1	linux1	linux1	Public	192.168.88.121	Fixed	DNS, files
Node 1 VIP	Node 1	Selected by Oracle Clusterware	linux1-vip	Virtual	192.168.88.21	Fixed	DNS, files
Node 1 Private	Node 1	linux1	linux1-priv	Private	192.168.198.21	Fixed	DNS, files, or none
Node 2 Public	Node 2	linux2	linux2	Public	192.168.88.122	Fixed	DNS, files
Node 2 VIP	Node 2	Selected by Oracle Clusterware	linux2-vip	Virtual	192.168.88.22	Fixed	DNS, files
Node 2 Private	Node 2	linux2	linux2-priv	Private	192.168.198.22	Fixed	DNS, files, or none
SCAN VIP 1	none	Selected by Oracle Clusterware	oracle-scan	virtual	192.168.88.200		DNS
SCAN VIP 2	none	Selected by Oracle Clusterware	oracle-scan	virtual	192.168.88.201		DNS
SCAN VIP 3	none	Selected by Oracle Clusterware	oracle-scan	virtual	192.168.88.202		DNS

A virtual host name<*public hostname*>-vip is a public node name that is used 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: Linux1-vip.

DNS zone/domain: linux.lii01.livun.vom

RAC database Load-Balancing, Failover, and High Availability

Oracle Clusterware increases availability by load balancing and failover, takes advantage of the redundant resources within a cluster database.

Clients that connect to any instance of the database use the SCAN name for the database. This entry also enables Oracle Enterprise Manager to discover an Oracle RAC database. Three SCAN IP addresses are configured for the cluster, and allocated to servers. If the chosen address fails, then the connection request to the SCAN name fails over to the next address. Thus, if an instance fails, then clients can still connect using another instance.

Clients are redirected by the SCAN Listener to a local listener on the least loaded node that is running the instance for a particular service. That local listener either redirects the client to a Dispatcher process (if the database was configured for Shared Server), or redirects the client to a dedicated server process. Local Listeners are configured to respond to database connection requests, and to non-database connection requests, such as external procedures or XDB requests.

To administer 11g release 2 (11.2) local and scan listeners using the lsnrctl command, set your `ORACLE_HOME` environment variable to the path for the grid infrastructure home (Grid home in this guide: /u01/app/cluster/product/11.2.0/grid).

TIPS: Do not attempt to use the lsnrctl commands from Oracle home locations for previous releases, as they cannot be used with the new release.

Before you install Oracle RAC, during the Oracle grid infrastructure installation, Oracle Net Configuration Assistant creates and starts a default listener in the Grid home called LISTENER. The listener is configured with default protocol listening addresses. The listener is configured to respond to connection requests that are directed to one protocol address specified during installation.

During Oracle RAC installation, the `Oracle RAC database uses the Oracle Clusterware listener`, and configures service information about the Oracle RAC database. An Oracle Database 11g release 2 (11.2) database service automatically registers its information with the listener, such as its `service name`, `instance names`, and `load information`. This feature, called service registration, does not require configuration in the listener.ora file. After listener creation, Oracle Net Configuration Assistant starts the listener.

Since Oracle Database 11g Release 2, SCAN is an essential part of the Oracle RAC database configuration and therefore the `REMOTE_LISTENER` parameter is set to the SCAN per default, which identifies the global list of listeners, assuming that the database is created using standard Oracle tools (e.g. the formerly

mentioned DBCA). This allows the instances to register with the SCAN Listeners as remote listeners to provide information on what services are being provided by the instance, the current load and a recommendation on how many incoming connections should be directed to the instance.

During Oracle Database creation, the `LOCAL_LISTENER` parameter is automatically configured to point to the local listener for the database. The Database Agent sets the `LOCAL_LISTENER` parameter to a connect descriptor that does not require a TNS alias.You can set a value manually for `LOCAL_LISTENER`. However, **Oracle recommends that you leave the parameter unset so that the Database Agent can maintain it automatically**. If you set `LOCAL_LISTENER`, then the Agent does not automatically update this value. If you do not set `LOCAL_LISTENER`, then the Database Agent automatically keeps the database associated with the Grid home's node listener updated, even as the ports or IP of that listener are changed.

If DBCA detects more than one listener on the node, then it displays a list of the listeners. You can select one or all of these listeners with which to register your database.

With Oracle Clusterware 11g release 2 and later, Database Configuration Agent (DBCA) no longer sets the `LOCAL_LISTENER` parameter. The Oracle Clusterware agent that starts the database sets the `LOCAL_LISTENER` parameter dynamically, and it sets it to the actual value, not an alias. So listener_alias entries are no longer needed in the tnsnames.ora file. For the `REMOTE_LISTENER` parameter, Oracle Clusterware uses the EZ connect syntax *scanname: scanport*, so no entries are needed for the `REMOTE_LISTENER` parameter in the tnsnames.ora file.

After installation, you can modify listeners by using **srvctl** commands, or by using **NetCA**. Manual editing of the `listener.ora` file is **not** required.

Listener Registration and PMON Discovery: When a listener starts after the Oracle instance starts, and the listener is listed for service registration, registration does not occur until the next time the Oracle Database process monitor (PMON) discovery routine starts. By default, PMON discovery occurs every 60 seconds.To override the 60-second delay; use the SQL ALTER SYSTEM REGISTER statement. This statement forces the PMON process to register the service immediately. Oracle recommends that you create a script to run this statement immediately after starting the listener. If you run this statement while the listener is up and the instance is already registered, or while the listener are down, then the statement has no effect.

Net Service Names (tnsnames.ora) is created on each node with **net service names.** A connect identifier is an identifier that maps to a connect descriptor. A connects descriptor contains the following information:

- The **network route to the service,** including the location of the listener through a protocol address
- The **SERVICE_NAME** for an Oracle Database

Net Service Names for Database Connections: The SERVICE_NAME parameter you use in the tnsnames.ora file is singular, because you can specify only one service name. There is also the **SERVICE_NAMES** parameter in the **database initialization file**. The SERVICE_NAMES parameter defaults the global database name, a name comprising the DB_NAME and DB_DOMAIN parameters in the initialization parameter file.

Net Service Name for Instance Connections: Clients that connect to a particular instance of the database use the SCAN name for the instance. This entry, for example, enables Oracle Enterprise Manager to discover the instances in the cluster. These entries are also used to start and stop instances.

Net Services Profile (sqlnet.ora): Oracle Universal Installer starts Oracle Net Configuration Assistant after the database is installed, and creates the Net Services profile, or *sqlnet.ora* file.

By default, the sqlnet.ora file is located in the following directory: *$ORACLE_HOME/network/admin*

In this directory, in addition to the default sqlnet.ora file generated during installation, you also can find a sample sqlnet.ora file in the directory sample.

During installation, Oracle Net Configuration Assistant creates the following entries in the sqlnet.ora file, where Oracle_base is the path to the Oracle RAC installation owner's Oracle base directory:

```
NAMES.DIRECTORY_PATH=(TNSNAMES, EZCONNECT)
ADR_BASE =Oracle_base
```

The parameter NAMES.DIRECTORY_PATH specifies the priority order of the naming methods to use to resolve connect identifiers to connect descriptors.

The Easy Connect naming method eliminates the need for service name lookup in the tnsnames.ora files for TCP/IP environments. With Easy Connect, clients use a connect string for a simple TCP/IP address, consisting of a host name and optional port and service name. If you use this method, then no naming or directory system is required.

Oracle storage elements under Grid structure

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 of Oracle Database files, and OCFS2 for Oracle Clusterware components.

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 (files). 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 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.

Note: If you choose not to use Oracle ASM for storing your Oracle Clusterware files, then both the voting disks and the OCR must reside on a cluster file system that you configure before you install Oracle Clusterware in the Grid home.

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

- Three Oracle Clusterware Registry (OCR) files, 300 MB each, or 900 MB total disk space
- Three voting disk files, 300 MB each, or 900 MB total disk space

If you are not using Oracle ASM for storing Oracle Clusterware files, then for best performance and protection, you should use multiple disks, each using a different disk controller for voting disk file placement. Ensure that each voting disk is configured so that it does not share any hardware device or have a single point of failure.

From Operating System administrator's view, there are three file types that make up an Oracle database, plus a few others that exist externally to the database and are, strictly speaking, optional.

Three required file types in a database: the controlfile, the online redo log files, and the datafiles.

- The controlfile stores integrity information of database and pointers to the rest of the database files.
- The online redo logs store recent change vectors applied to the database.
- The datafiles store the data.

External files include

- The parameter file (pfile, spfile), which is used to start oracle instance running environment;
- The password file, which is used to authenticate user before database is opened;
- Archive redo logs, and the log and trace files.

- /etc/oratab Configuration for Oracle RAC: Oracle creates an entry for each Oracle RAC database in the /etc/oratab configuration file. The oratab file is created by the root.sh script during installation, and it is updated by the Database Configuration Assistant when creating or deleting a database. The oratab file entry is also created automatically by the Database Agent when a database is first started on a node where it has not run previously. Oracle ASM Agent creates Oracle ASM oratab entries. Oracle Enterprise Manager uses the oratab file during service discovery to determine the name of the Oracle RAC database, and to determine if the database should be started automatically when the system is restarted. The database entry has the following syntax: $DB_UNIQUE_NAME:$ORACLE_HOME:N, A colon, ':', is used as the field terminator. A new line terminates the entry. Lines beginning with a pound sign, '#', are comments. Because all the instances of an Oracle RAC database have the same DB_UNIQUE_NAME, but each instance has its own ORACLE_SID, we use the $DB_UNIQUE_NAME environment variable in the oratab file as the database entry. The $DB_UNIQUE_NAME identifier for your RAC database must be unique across your enterprise. $ORACLE_HOME is the directory path to the database, and N indicates that the database should not be started at restart time. The following is an example entry for a database named sales: sales:/u01/app/oracle/sales:N

Oracle Database logic components

For both single-instance and cluster database environments, an Oracle Database is divided into smaller logical areas of space known as `tablespaces`. Each tablespace corresponds to `one or more data files` on the shared storage.

- SYSTEM: Consists of the data dictionary, including definitions of tables, views, and stored procedures needed by the database. Oracle Database automatically maintains information in this tablespace.
- SYSAUX: An auxiliary system tablespace that contains the DRSYS (contains data for Oracle Text), CWMLITE (contains the OLAP schemas), XDB (for XML features), ODM (for Oracle Data Mining), TOOLS (contains Oracle Enterprise Manager Tables), INDEX, EXAMPLE, and OEM-REPO tablespaces.
- USERS: Consists of application data. As you create and enter data into tables, Oracle Database fills this space with your data.
- TEMP: Contains temporary tables and indexes created during SQL statement processing. You may need to expand this tablespace if you are running a SQL statement that involves significant sorting, such as ANALYZE COMPUTE STATISTICS on a very large table, or the constructs GROUP BY, ORDER BY, or DISTINCT.
- Logical data storage (segments) is abstracted from physical data storage (datafiles) by tablespaces.
- Segments consist of multiple extents, which consist of multiple Oracle blocks, which consist of one or more operating system blocks. A segment can have extents in several datafiles.
- Redo Log Files: Each instance is configured with at least two redo log files that are stored in the shared storage. If you use a file system, then these files are shared file system files. If you use Oracle ASM, then these files are stored on the Oracle ASM disk group. The file names of the redo log files that are created with the preconfigured database configuration options vary by storage type. Unless you are using a cluster file system, you must enter the block or raw device names.

- UNDOTBSn: Managing Undo Tablespaces in Oracle RAC. Oracle Database stores rollback or undo information in undo tablespaces. Oracle recommends that you use automatic undo management. Automatic undo management is an automated undo tablespace management mode that is easier to administer than manual undo management. When Oracle ASM and Oracle Managed Files are used along with Automatic Undo Management, an instance that is started for the first time, and thus does not have an undo tablespace, will have its undo tablespace created for it by another instance automatically. The same is also true for redo logs.
- RBS: If you do not use automatic undo management, then Oracle Database uses the RBS tablespace for the rollback segments.

Some tablespaces (data files) are critical to the database remaining open: the files that make up the SYSTEM tablespace, the active UNDO tablespace, and the controlfile copies. Damage to any of these will cause the database to crash. Some files are critical for performance: the online redo log files and the controlfiles. I/O on these can be a serious bottleneck. By considering the characteristics of each file, a RAID strategy should become apparent. For example, the SYSTEM and UNDO tablespaces should be on RAID 1 volumes, so that they will always be available. The online redo logs are protected by multiplexing, so you don't have to worry about hardware fault tolerance, but performance is vital, so put them on RAID 0 volumes. The controlfile copies are critical for performance, but if any copy is damaged, the database will crash; thus, RAID 0+1 could be appropriate. Your other datafiles could perhaps go on RAID 5 volumes, unless they are particularly important or volatile.

Prepare Guest Nodes and Operating System

Node Hardware environment

Each node you make part of your cluster, or Oracle Clusterware and Oracle RAC installation, must satisfy the minimum hardware requirements of the software. These hardware requirements can be categorized as follows:

- Physical memory (at least 1.5 gigabyte (1576 MB) of RAM)
- An amount of swap space equal to the amount of RAM
- 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
- A minimum of 1024 x 786 display resolution, so that Oracle Universal Installer (OUI) displays correctly
- All servers that are used in the cluster must have the same chip architecture, for example, all 32-bit processors or all 64-bit processors
- Adequate disk space (about 10 GB) in the software installation locations to store the Oracle software. You need at least 5.5 GB of available disk space for the Grid home directory, which includes both the binary files for Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM) and their associated log files, and at least 4 GB of available disk space for the Oracle Database home directory, or Oracle home directory

VMWare Virtual Network adapters on laptop is a bright between Laptop physical network and VM hosts virtual network, findout or set host-only network interface `VMnet1` IP address `192.168.198.1`, and NAT network interface `VMnet8` IP address `192.168.88.1`

Note: When you install Oracle software, Oracle Universal Installer (OUI) automatically performs hardware prerequisite checks and notifies you if they are not met.Refer to the *Oracle Grid Infrastructure Installation 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.

First Virtual Machine initial feature

Memory: `1.6GB (1576MB) or more`

Processors: `1 or more`

Storage: `20G (25G will be preferred) Hard Disks under C:\vmware\Linux1`

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

- The operating system version
- 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 version that is not supported by Oracle Database 11*g* release 2 (11.2), then you must first upgrade your operating system before installing Oracle Real Application Clusters 11*g*.

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.

First Node

Use VMWare Workstation 10 Wizard, Easy install, to create the virtual machine and start OS installation, Steps should be very straight forward:

OS Downloaded to local: C:\Users\lchen\Downloads\RedHat5.8\ rhel-server-5.8-x86_64-dvd.iso

Virtual Machine Name: Linux1;

Location: C:\Vmware\Linux1

Specify 20G Disk drive for root filesystem installation; it will also hold Oracle production softwares.

Start Linux OS installation by Easy install

Linux OS will start after installation, and login as root

Kernel parameters and packages for Oracle

Add Swap Size if needed, example:

To add a swap partion:

```
1. Use fdisk or parted to create a new disk parttion, /dev/sda3, select partition type as linux swap;
2. Use mkswap /dev/sda3, and swapon /dev/sda3
3. To enable this new swap partition at boot time, edit /etc/fstab to include /dev/sda3
```

To add a swap file:

```
1. dd if=/dev/zero of=/swapfile bs=250MB count=2
2. mkswap /swapfile; swapon /swapfile
3. To enable this new swap partition at boot time, edit /etc/fstab to include /swapfile
   /swapfile               swap                    swap    defaults        0 0
```

Modify Kernel parameters

To edit /etc/sysctl.conf, add require OS kernel parameters :

```
# Kernel sysctl configuration file for Red Hat Linux
kernel.shmall = 2097152
kernel.shmmax = 2147483648
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.file-max = 68157144
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 4194304
net.core.rmem_max = 4194304
net.core.wmem_default = 1048576
net.core.wmem_max = 1048576
fs.aio-max-nr = 1048576
```

Then execute the following command for change to take effect:

```
# /sbin/sysctl -p
```

Modity oracle users resource limits

To improve the performance of the software on Linux systems, you must increase the following resource limits for the Oracle software owner users (grid, oracle), Change `/etc/security/limits.conf` file with a root user and add following lines:

```
grid soft nproc 2047
grid hard nproc 16384
grid soft nofile 1024
grid hard nofile 65536
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
```

Add following entry in `/etc/pam.d/login`

```
session    required     pam_limits.so
```

Add lines similar to the following to the `/etc/profile`, make the following changes to the default shell startup file, to change ulimit setting for all Oracle installation owners

```
if [ $USER = "oracle" ] || [ $USER = "grid" ]; then
        if [ $SHELL = "/bin/ksh" ]; then
              ulimit -p 16384
              ulimit -n 65536
        else
              ulimit -u 16384 -n 65536
        fi
        umask 022
fi
```

Oracle Grid Infrastructure and Oracle RAC Package

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 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
```

To install packages:

```
# rpm -ivh 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 (Red Hat Enterprise Linux 5) or download the required package version from your Linux vendor's Web site.

Packages required for Red Hat Enterprise Linux 5 (x86-64)

```
binutils-2.17.50.0.6
compat-libstdc++-33-3.2.3
compat-libstdc++-33-3.2.3 (32 bit)
elfutils-libelf-0.125
elfutils-libelf-devel-0.125
gcc-4.1.2
gcc-c++-4.1.2
glibc-2.5-24
glibc-2.5-24 (32 bit)
glibc-common-2.5
glibc-devel-2.5
glibc-devel-2.5 (32 bit)
glibc-headers-2.5
ksh-20060214
libaio-0.3.106
```

```
libaio-0.3.106 (32 bit)
libaio-devel-0.3.106
libaio-devel-0.3.106 (32 bit)
libgcc-4.1.2
libgcc-4.1.2 (32 bit)
libstdc++-4.1.2
libstdc++-4.1.2 (32 bit)
libstdc++-devel 4.1.2
make-3.81
sysstat-7.0.2
unixODBC-2.2.11
unixODBC-2.2.11 (32 bit)
unixODBC-devel-2.2.11
unixODBC-devel-2.2.11 (32 bit)
```

ocfs2 packages for Oracle Cluster File System (OCFS2)

```
ocfs2-2.6.18-308.el5-1.4.7-1.el5.x86_64.rpm
ocfs2-tools-1.6.3-2.el5.x86_64.rpm
ocfs2-2.6.18-308.el5-debuginfo-1.4.7-1.el5.x86_64.rpm
ocfs2-tools-debuginfo-1.6.3-2.el5.x86_64.rpm
ocfs2-tools-devel-1.6.3-2.el5.x86_64.rpm
ocfs2console-1.6.3-2.el5.x86_64.rpm
```

ASMLib packages for ASM

```
oracleasm-support-version.arch.rpm
oracleasm-kernel-version.arch.rpm
oracleasmlib-version.arch.rpm
```

To find if you have an existing version of the cvuqdisk package:

rpm -qi cvuqdisk

You can also use either ***up2date or YUM (Yellow dog Updater Modified)*** to install packages and their dependencies on some Linux systems.

To make YUM work, first moutn DVD image, then modify YUM repository location

```
#mount /dev/hda /mnt; YUM repository location: file:///mnt/Server
```

For some packages you downloaded to your laptop, you need to share them among your PC and linuix servers, install package samba client program (`samba3x-client`) on linux,

```
# mount //winserver/software /install -o username=lrchen
```

winserver is your laptop ip (192.168.10.100, use IP address if you have problem), and /software is your shared directory on your laptop, lrchen is the user name on your laptop; /install is mountpoint on linux server.

Start nfs service on linux1,

```
# exportfs -o async linux2:/software  (for stoping nfs export: # exportfs -u linux2:/software)
```

On linux2:

```
# showmount -e linux1; mount linux1:/softare /mnt; cd /mnt; rpm -i ocfs*
```

Setting the Time synchronize

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.

If you use NTP, then you must configure it using the `-x` flag. If the -x option is included on the command line, the clock will never be stepped and only slew corrections will be used.

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

# /etc/init.d/ntpd start
```

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

Set Network environment

We recommend you to disable Firewall Option and SELinux before install oracle RAC.

/etc/resolv.conf is responsible for DNS resolvation, the total timeout period configured—a combination of options attempts (retries) and options timeout (exponential backoff)—should be less than 30 seconds. For example:

```
[root@linux1 ~]# cat /etc/resolv.conf
    ; generated by /sbin/dhclient-script
    search localdomain
    nameserver 10.168.88.1
```

System (NIS) can cause problems with Oracle SCAN address resolution. Oracle recommends that you place the nis entry at the end of the search list. For example:

```
    hosts:    files   dns   nis
```

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 (TIPs: using **ip lable short name** and also **FQDN**, otherwise some services/applications will be stucked when starting)

```
root@linux1 ~]# cat /etc/hosts

    # Do not remove the following line, or various programs
    # that require network functionality will fail.
    127.0.0.1               localhost.localdomain localhost
    ::1             localhost6.localdomain6 localhost6

    192.168.88.121  linux1.localdomain linux1
    192.168.88.21   linux1-vip.localdomain linux1-vip
    192.168.198.21  linux1-priv.localdomain linux1-priv

    192.168.88.122  linux2.localdomain linux2
    192.168.88.22   linux2-vip.localdomain linux2-vip
    192.168.198.22  linux2-priv.localdomain linux2-priv
```

Create Oracle Users and Groups

To install the Oracle Grid Infrastructure for a cluster software and Oracle RAC, you must create the following operating system groups and users:

- The Oracle Inventory group (typically, ***oinstall***) for all installations. The Oracle Inventory group must be the primary group for Oracle software installation owners. 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.
- The OSDBA group (typically, ***dba***) for Oracle Database authentication. The OSDBA group is a system privileges group whose members are granted the SYSDBA 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".
- An Oracle software owner. 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 the Oracle Grid Infrastructure for a cluster and the Oracle RAC software using separate software owners to separate Oracle Grid Infrastructure for a cluster administrative privilege from Oracle Database administrative privileges, then typically you would use ***grid*** for the Oracle Grid Infrastructure for a cluster software owner and ***oracle*** for the Oracle RAC software owner. 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` privileges for the databases associated with their OSDBA 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 the next section, 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 software installation can be owned by only one user. You cannot have one user that owns the Oracle Clusterware software installation and a different user that owns the Oracle ASM software installation, here is user ***grid***.

The optional users and groups you can create are:

- The Oracle Automatic Storage Management Group or OSASM group (typically `asmadmin`). Create this group as a separate group if you want to have separate administration privilege groups for Oracle ASM and Oracle Database administrators. Members of the OSASM group can use SQL to connect to an Oracle ASM instance as `SYSASM` using operating system authentication. The `SYSASM` privileges permit mounting and dismounting disk groups, and other storage administration tasks. `SYSASM` privileges do not provide access privileges to an Oracle Database instance. If you do not create a separate OSASM group, then the OSDBA group (`dba`) is the OSASM group.
- The ASM Database Administrator group (OSDBA for ASM group, typically `asmdba`). Members of the OSDBA group for Oracle ASM are granted read and write access to files managed by Oracle ASM. The Oracle Grid Infrastructure for a cluster installation owner and all Oracle Database software owners (for example, `oracle`) must be a member of this group, and all users with OSDBA membership for databases that require access to the files managed by Oracle ASM should be members of the OSDBA for ASM group.
- The OSOPER for Oracle Database group (typically, `oper`). Create this group if you want certain operating system users to have a limited set of database administrative privileges (the `SYSOPER` privilege). Members of the OSDBA group automatically have all privileges granted by the `SYSOPER` privilege.
- The OSOPER for Oracle ASM group (typically `asmoper`). Members of this group are granted access to a subset of the `SYSASM` privileges, such as starting and stopping the Oracle ASM instance.

Now let's start the prerequisite steps for Oracle installation. For this we'll create 2 groups: oinstall & dba, 2 users: oracle & grid

```
groupadd -g 501 oinstall
groupadd -g 502 dba
useradd -u 501 -g oinstall -G dba grid
useradd -u 502 -g oinstall -G dba oracle
```

Note: If installing Oracle RAC on Microsoft Windows, then OUI automatically creates the `ORA_DBA` group for authenticating SYSDBA access. Also, if you install the Oracle RAC software while logged in to an account with administrative privileges, then you do not have to create a separate user for the installation.

Oracle Flexible Architecture

The Oracle Home will need a file system into which it can be installed. Oracle Corporation has designed OFA, the **Optimal Flexible Architecture**, as a file system directory structure that should make maintaining multiple versions of multiple Oracle products straightforward. The heart of OFA is two environment variables: **ORACLE_BASE** and **ORACLE_HOME**. The ORACLE_BASE directory is one directory on the server, beneath which all the Oracle software (all products, all versions) should be installed. Each version of each product will then have its own ORACLE_HOME, beneath the ORACLE_BASE. This structure should ensure that many databases can be created and upgraded without ever ending up with files in inappropriate locations

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 traces files from installations of Oracle software. These files are also copied to the respective Oracle homes for future reference

The `oraInventory` directory is created by OUI during installation.By default, the Oracle Inventory directory is not installed under the Oracle Base directory. OUI creates the Oracle Inventory directory in the path $ORACLE_BASE/../oraInventory. For example, if ORACLE_BASE is set to /u0/app/grid, then the Oracle Inventory directory is created in the path /u0/app/oraInventory, so that the central inventory for all installations is outside of the Oracle base for this particular Oracle installation user. 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.

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
```

2. 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`).

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

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

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.

You also use the local file system for the Oracle home directory, if you want to install a different version of Oracle RAC or Oracle Database on the same server, then you must use a separate Oracle home directory for each software installation. Multiple versions 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.

```
mkdir -p /u01/app/cluster/product/11.2.0/grid/
chown -R grid:oinstall /u01
mkdir /u01/app/oracle
chown oracle:oinstall /u01/app/oracle
chmod -R 775 /u01/
```

After running these commands, you have the following groups and users:

- An Oracle central inventory group or oraInventory group (oinstall). Members, who have the central inventory group as their primary group, are granted the OINSTALL permission to write to the oraInventory directory.
- A single system privileges group that is used as the OSASM, OSDBA, OSDBA for ASM, and OSOPER for ASM group (dba), whose members are granted the SYSASM and SYSDBA privilege to administer Oracle Clusterware, Oracle ASM, and Oracle Database, and are granted SYSASM and OSOPER for ASM access to the Oracle ASM storage.
- An Oracle grid installation for a cluster owner (grid), with the oraInventory group as its primary group, and with the OSASM group as the secondary group, with its Oracle base directory /u01/app/grid.
- An Oracle Database owner (oracle) with the oraInventory group as its primary group, and the OSDBA group as its secondary group, with its Oracle base directory /u01/app/oracle.
- /u01/app owned by grid:oinstall with 775 permissions before installation, and by root after the root.sh script is run during installation. This ownership and permissions enables OUI to create the Oracle Inventory directory, in the path /u01/app/oraInventory.
- /u01 owned by grid:oinstall before installation, and by root after the root.sh script is run during installation.
- /u01/app/cluster/product/11.2.0/grid owned by grid:oinstall with 775 permissions. These permissions are required for installation, and are changed during the installation process.
- /u01/app/grid owned by grid:oinstall with 775 permissions before installation, and 755 permissions after installation.
- /u01/app/oracle owned by oracle:oinstall with 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.

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.

Note: 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.

User Environment varibles

Variables can be set at various levels with various degrees of persistence on a Windows system, ranging from permanent, system-wide variables set in the Windows registry to variables set interactively within a command shell. As a general rule, variables set at a higher level (such as within the registry) can be overruled at a lower level (such as within a shell). The highest level for variables is in the registry

Refer to the *Oracle Grid Infrastructure Installation Guide* for your platform for more information on how to configure the Oracle software owner environment before installation.

On Oracle Linux, you run Oracle Universal Installer (OUI) from the ***grid*** (or ***oracle***) account. OUI obtains information from the environment variables configured for the user. Before running OUI, you must make the following changes to the Oracle Grid Infrastructure for a cluster software owner's shell startup file:

- Set the default file mode creation mask (`umask`) of the installation user 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.
- 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.

To connect to an ASM instance with SQL*Plus, set the ORACLE_SID environment variable to the ASM SID. The default ASM SID for a single-instance database is +ASM, and the default SID for ASM for an Oracle RAC node is +ASMnode_number where node_number is the number of the node. Depending on your operating system and whether you installed ASM in a separate ASM home, you might have to change other environment variables.

After you have saved your changes, run the shell startup script to configure the environment. For user ***grid***, Change the ~/ .bash_profile (and .bashrc) file and add the following lines:

[grid@linux1 ~]$ cat .bash_profile

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

#export PATH
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/cluster/product/11.2.0/grid
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export
PATH=$PATH:$HOME/bin:$ORACLE_HOME/bin:/bin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/X11R6/bin
umask 022
```

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.

Establish user equivalence with SSH

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:

```
[root@linux1 ~]# su - grid
[grid@linux1 ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/grid/.ssh/id_rsa):
Created directory '/home/grid/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/grid/.ssh/id_rsa.
Your public key has been saved in /home/grid/.ssh/id_rsa.pub.
The key fingerprint is:
25:e9:f9:14:a4:b0:cc:d8:02:db:99:02:a2:93:de:63 grid@linux1
[grid@linux1 ~]$
```

- 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).

Practice: Oracle Universal Installer installs the binaries in one node and then propagates the files to the other nodes. For this, it uses ssh and scp command in the background during installation to run remote commands and copy files to the other cluster nodes. So SSH must be configured so that these commands not prompt for a password. For this power on the first machine, login with root user, switch to the oracle user and generate RSA and DSA key pairs

```
# su - oracle ; (same process as $su - grid)
$ ssh-keygen -t rsa (click Enter)
$ ssh-keygen -t dsa (click Enter)

$ cat ~/.ssh/id_rsa.pub > ~/.ssh/authorized_keys
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

issue # ssh linux1 to test

Please make sure that after running above commands on each node, you're not prompted for the password for the second time. To enable SSH user equivalency:

1. Start the SSH agent and load the SSH keys into memory using the following commands:

```
$ /usr/bin/ssh-agent $SHELL
$ /usr/bin/ssh-add
Enter passphrase for /home/oracle/.ssh/id_rsa
Identity added: /home/oracle/.ssh/id_rsa (/home/oracle/.ssh/id_rsa)
Identity added: /home/oracle/.ssh/id_dsa (/home/oracle/.ssh/id_dsa)
```

These commands start the ssh-agent on the local node, and load the RSA and DSA keys into the current session's memory if you set passphrase locally, so that you are not prompted to use pass phrases when issuing SSH commands.

2. To test if you have configured SSH correctly, run the following command. If you have configured SSH correctly, then you are not prompted for a password or a pass phrase.

```
$ ssh linux2 date
```

Clone Second Node

After you have verified that your system meets the basic requirements for installing Oracle RAC, the next step is to prepare another node in Oracle RAC environment.

Practice: Using Linked Clones

A linked clone is a copy of a virtual machine that shares virtual disks with the parent virtual machine in an ongoing manner.

Because a linked clone is made from a snapshot of the parent, disk space is conserved and multiple virtual machines can use the same software installation. All files available on the parent at the moment you take the snapshot continue to remain available to the linked clone.

Ongoing changes to the virtual disk of the parent do not affect the linked clone, and changes to the disk of the linked clone do not affect the parent. A linked clone must have access to the parent. Without access to the parent, you cannot use a linked clone.

Another approach of deplicating another Server Linux2 with Linux1, just shutdown virtual machine Linux1 and then copy c:\VMWare\Linux1 to c:\VMWare\Linux2

Customize Both Nodes for Oracle Grid Infrastracture

For both Node linux1 & linux2, Power off virtual machine, build storage and network according to Oracle RAC requirement.

Create RAC share storage under c:\VMware\LlinuxStorage.

For linux1, we create new disk, for linux2, we using exist disk (created by linux1) to create hard disks.

Name the new hard drive to "`ocfs2disk.vmdk`" under a separate folder "C:\vmware\LinuxStorage". This disk driver is for Oracle Cluster components of voting disk and OCR

After creating the first device, create more three devices with 3-4GB in size (***asmdisk1.vmdk, asmdisk2.vmdk, asmdisk3.vmdk***) within the same directory and make all of them "Independent->Persistent" and don't allocate the disk space for each of them; These disk dirves are for Oracle ASM storage.

Change the device node for each of these new created disks. Start from the firstly added hard drive, select it, click on advanced button and make it SCSI 1:0. For next hard drive make it 1:1 and so on

Add new Ethernet device on virtual machine:

. NAT: Used to share the host's IP address, used by Oracle Gird Infrastructure as public IP address.

. Host-only: A private network shared the host, used by Oracle Grid Infrastructure as private IP address.

Create **2 Ethernet adapters**, each for NAT/Host-only

Final Devices configuration should looks like:

Important: After all these new hardware building finished, locate the configuration file of the virtual machine and edit it (C:\VMware\Linux1\Linux1.vmx; C:\VMware\Linux2\Linux2.vmx)

```
.encoding = "windows-1252"
config.version = "8"
virtualHW.version = "7"
vcpu.hotadd = "TRUE"
scsi0.present = "TRUE"
scsi0.virtualDev = "lsilogic"
memsize = "1800"
mem.hotadd = "TRUE"
scsi0:0.present = "TRUE"
scsi0:0.fileName = "Linux1.vmdk"
ide1:0.present = "TRUE"
ide1:0.fileName = "auto detect"
ide1:0.deviceType = "cdrom-raw"
ethernet0.present = "TRUE"
ethernet0.connectionType = "nat"
ethernet0.virtualDev = "e1000"
ethernet0.wakeOnPcktRcv = "FALSE"
ethernet0.addressType = "generated"
usb.present = "TRUE"
ehci.present = "TRUE"
ehci.pciSlotNumber = "35"
sound.present = "TRUE"
sound.fileName = "-1"
sound.autodetect = "TRUE"
serial0.present = "TRUE"
serial0.fileType = "thinprint"
pciBridge0.present = "TRUE"
pciBridge4.present = "TRUE"
pciBridge4.virtualDev = "pcieRootPort"
pciBridge4.functions = "8"
pciBridge5.present = "TRUE"
pciBridge5.virtualDev = "pcieRootPort"
pciBridge5.functions = "8"
pciBridge6.present = "TRUE"
pciBridge6.virtualDev = "pcieRootPort"
pciBridge6.functions = "8"
pciBridge7.present = "TRUE"
pciBridge7.virtualDev = "pcieRootPort"
```

```
pciBridge7.functions = "8"
vmci0.present = "TRUE"
usb.vbluetooth.startConnected = "TRUE"
displayName = "Linux1"
guestOS = "rhel5-64"
nvram = "Linux1.nvram"
virtualHW.productCompatibility = "hosted"
powerType.powerOff = "soft"
powerType.powerOn = "soft"
powerType.suspend = "soft"
powerType.reset = "soft"
extendedConfigFile = "Linux1.vmxf"
uuid.bios = "56 4d ef 43 98 08 ca 21-35 a5 58 60 18 c5 ab 82"
uuid.location = "56 4d ef 43 98 08 ca 21-35 a5 58 60 18 c5 ab 82"
replay.supported = "FALSE"
replay.filename = ""
scsi0:0.redo = ""
pciBridge0.pciSlotNumber = "17"
pciBridge4.pciSlotNumber = "21"
pciBridge5.pciSlotNumber = "22"
pciBridge6.pciSlotNumber = "23"
pciBridge7.pciSlotNumber = "24"
scsi0.pciSlotNumber = "16"
usb.pciSlotNumber = "32"
ethernet0.pciSlotNumber = "33"
sound.pciSlotNumber = "34"
vmci0.pciSlotNumber = "36"
ethernet0.generatedAddress = "00:0c:29:c5:ab:82"
ethernet0.generatedAddressOffset = "0"
vmci0.id = "415607682"
vmotion.checkpointFBSize = "33554432"
cleanShutdown = "TRUE"
softPowerOff = "FALSE"
usb:1.speed = "2"
usb:1.present = "TRUE"
usb:1.deviceType = "hub"
usb:1.port = "1"
usb:1.parent = "-1"
ide1:0.autodetect = "TRUE"
tools.syncTime = "FALSE"
tools.remindInstall = "FALSE"
toolsInstallManager.updateCounter = "1"
unity.wasCapable = "TRUE"
disk.locking = "FALSE"
diskLib.dataCacheMaxSize = "0"
scsi1.present = "TRUE"
scsi1.virtualDev = "lsilogic"
scsi1.sharedBus = "virtual"
scsi1:0.present = "TRUE"
scsi1:0.fileName = "C:\VMware\LinuxStorage\ocfs2disk.vmdk"
scsi1:0.deviceType = "disk"
scsi1:1.present = "TRUE"
scsi1:1.fileName = "C:\VMware\LinuxStorage\asmdisk1.vmdk"
scsi1:1.deviceType = "disk"
scsi1:2.present = "TRUE"
scsi1:2.fileName = "C:\VMware\LinuxStorage\asmdisk2.vmdk"
scsi1:2.deviceType = "disk"
scsi1:3.present = "TRUE"
scsi1:3.fileName = "C:\VMware\LinuxStorage\asmdisk3.vmdk"
scsi1:3.deviceType = "disk"
ethernet1.present = "TRUE"
ethernet1.connectionType = "hostonly"
ethernet1.virtualDev = "e1000"
ethernet1.wakeOnPcktRcv = "FALSE"
ethernet1.addressType = "generated"
ethernet2.present = "TRUE"
ethernet2.virtualDev = "e1000"
ethernet2.wakeOnPcktRcv = "FALSE"
ethernet2.addressType = "generated"
ide0:0.present = "FALSE"
floppy0.present = "FALSE"
```

Add the lines that are marked in bold/red to the configuration file to make the devices be shared between nodes

- Specify **diskLib.dataCacheMaxSize** = “0″ to turn off the disk caching for clustered virtual machines.
- By specifying **disk.locking =** “FALSE” will allow any virtual machine to load a SCSI disk device even it’s in use by an another virtual machine
- By specifying **scsi1.sharedBus** = “virtual” will give the whole bus the ability to be shared. This prevents the locking of this specific disk

Option: To synchronize the time on the virtual machine with the host machine execute “vmware-toolbox” on the terminal window and check the check box

Option: Edit /boot/grub/grub.conf file and add “clock=pit nosmp noapic nolapic” to the line that reads kernel /boot; the “clock=pit” prevents the clock for running too quickly and “nosmp noapic nolapic” prevents the clock from running too slowly.

After you make the change, reboot the machine for change to take effect.

Install Oracle Grid Infrastructure

For DNS setting, it will depend on your company network environment. Open Network Configuration and change IP addresses of each Ethernet device. Change hostname to linux1 and linux2 and assgin IPs:

Linux1

eth0: 192.168.88.121(linux1)

eth1: 192.168.198.21(linix1-priv)

Linux2

eth0: 192.168.88.122(linux2)

eth1: 192.168.198.22(linix2-priv)

IP label for linux1-vip & linux2-vip will be assigned by oracle cluster later.

Shared Storage

For all installations, you must choose the storage option to use for Oracle Clusterware files 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. 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, OCFS for Microsoft Windows, or General Parallel File System (GPFS) on IBM platforms
- Network file system (NFS), which is not supported on Linux on POWER or IBM zSeries Based Linux
- (Upgrades only) Shared disk partitions consisting of block devices or raw 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.

Storage Option	OCR and Voting Disk Files	Oracle Clusterware binaries	Oracle RAC binaries	Oracle Database Files	Oracle Recovery Files
Oracle Automatic Storage Management (Oracle ASM)	Yes	No	No	Yes	Yes
Oracle Automatic Storage Management Cluster File System (Oracle ACFS)	No	No	Yes	No	No
Local file system	No	Yes	Yes	No	No
NFS file system on a certified NAS	Yes	Yes	Yes	Yes	Yes
Shared disk partitions (block devices or raw devices)	No	No	No	No	No

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).

Shared OCFS2 storage for Oracle Cluster files

The examples in this guide use OCFS2 to store the Oracle Clusterware files (normally we use Oracle ASM to store the Oracle Clusterware and Oracle Database 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 this, run "**ocfs2console**" with a root user from the first node. Select Configure Nodes, click Add button and apply both nodes:

```
Linux1
192.168.88.121
7777

Linux2
192.168.88.122
7777
```

Then propagate the configuration to the second node. For this select "Propagate Configuration" from the Cluster menu.

```
Nodes configuration information located: /etc/ocfs2/cluster.conf, test node connection:
# nc -z linux1 7777; nc -z linux2 7777
```

Open port: 7777 if you forget disable Firewall setting on your linux server:

To configure O2CB to start at the boot unload and configure it on both nodes as a root user:

```
[root@linux1 ~]# /etc/init.d/o2cb unload
Stopping O2CB cluster ocfs2: OK
Unmounting ocfs2_dlmfs filesystem: OK
Unloading module "ocfs2_dlmfs": OK
Unmounting configfs filesystem: OK
Unloading module "configfs": OK

[root@linux1 ~]# /etc/init.d/o2cb configure
Configuring the O2CB driver.
```

```
This will configure the on-boot properties of the O2CB driver.
The following questions will determine whether the driver is loaded on
boot.  The current values will be shown in brackets ('[]').  Hitting
<ENTER> without typing an answer will keep that current value.  Ctrl-C
will abort.

Load O2CB driver on boot (y/n) [n]: y
Cluster stack backing O2CB [o2cb]:
Cluster to start on boot (Enter "none" to clear) [ocfs2]:
Specify heartbeat dead threshold (>=7) [31]:
Specify network idle timeout in ms (>=5000) [30000]:
Specify network keepalive delay in ms (>=1000) [2000]:
Specify network reconnect delay in ms (>=2000) [2000]:
Writing O2CB configuration: OK
Loading filesystem "configfs": OK
Mounting configfs filesystem at /sys/kernel/config: OK
Loading filesystem "ocfs2_dlmfs": OK
Mounting ocfs2_dlmfs filesystem at /dlm: OK
Starting O2CB cluster ocfs2: OK
```

Creating new disk partitions for OCFS2,

```
# fdisk /dev/sdb
Type n (to create new partition)
Type p to create a primary partition
Type 1 to provide the partition number
Double click on Enter and type "w" to save the changes
```

Now format the file system on the first node (Linux1). For this run ocfs2console program, select Format from the Tasks menu and click OK to format the drive. Press Ctrl+Q to quit

Now reboot both nodes, execute the following command on both nodes to mount the files system

```
# mount -t ocfs2 -o datavolume,nointr /dev/sdb1 /ocfs
```

And add the following line to the /etc/fstab to mount the files system on boot, for disk label information for Linux /dev/disk/by-label

```
LABEL=ocfs2 /ocfs ocfs2   _netdev,datavolume,nointr 0 0
```

Create a "clusterware" directory under /ocfs folder and change the owner:

```
# mkdir /ocfs/storage
# chown -R grid:oinstall /ocfs
```

Now to test the shared device, create a file in the /ocfs directory from the first node (rac1) and check the same folder in the second node.

```
# cd /ocfs; touch test_file
```

Shared ASMLib storage for Oracle Database files

Another option for configuring shared disks is to use the ASMLib utility. The following sections describe how to install and configure ASMLib, and how to use ASMLib to configure your shared disk devices:

- `Installing ASMLib`
- `Configuring ASMLib`
- `Using ASMLib to Create Oracle ASM Disks`

The Oracle Automatic Storage Management (`Oracle ASM)` library driver (`ASMLIB`) simplifies the configuration and management of the disk devices by eliminating the need to rebind disk devices used with Oracle ASM each time the system is restarted.

Without ASMLIB Linux 2.6 kernel and later, block device paths do not maintain permissions and path persistence unless you create a permissions or rules file on each cluster member node; block device paths that were /dev/sda can appear as /dev/sdb after a system restart. Adding new disks requires you to modify the udev file to provide permissions and path persistence for the new disk.

With ASMLIB, you define the range of disks you want to have made available as Oracle ASM disks. ASMLIB maintains permissions and disk labels that are persistent on the storage device, so that label is available even after an operating system upgrade. You can update storage paths on all cluster member nodes by running one oracleasm command on each node.

If you intend to use Oracle ASM on block devices for database storage for Linux, then Oracle recommends that you install the ASMLIB driver and associated utilities, and use them to configure the disks for Oracle ASM.

The ASMLib software is available from the Oracle Technology Network. Select the link for your platform on the ASMLib download page at:
http://www.oracle.com/technetwork/topics/linux/asmlib/index-101839.html

You should see four to six packages for your Linux platform. The oracleasmlib package provides the actual Oracle ASM library. The oracleasm-support package provides the utilities used to get the Oracle ASM driver up and running. Both of these packages must be installed.

The remaining packages provide the kernel driver for the Oracle ASM library. Each package provides the driver for a different kernel. You must install the appropriate package for the kernel you run. Use the uname -rm command to determine the version of the kernel on your server. The oracleasm kernel driver package has that version string in its name.

Note: The Oracle ASMLib kernel driver (`oracleasm`) is included with Oracle Linux 5. No driver package needs to be installed when using this kernel. The oracleasm-support and oracleasmlib packages still need to be installed.

Now that the `ASMLib` software is installed, a few steps have to be taken by the system administrator to make the Oracle ASM driver available. The Oracle ASM driver must be loaded, and the driver file system must be mounted. This is taken care of by the initialization script, /usr/sbin/oracleasm.

1. As the `root` user, run the following command:

```
# /usr/sbin/oracleasm configure -i
Configuring the Oracle ASM library driver
This will configure the on-boot properties of the Oracle ASM library driver.  The following questions will determine whether the driver is loaded on boot and what permissions it will have.  The current values will be shown in brackets ('[]').  Hitting <ENTER> without typing an answer will keep that current value. Ctrl-C will abort.

Default user to own the driver interface []: grid
Default group to own the driver interface []: oinstall
Start Oracle ASM library driver on boot (y/n) [n]: y
Scan for Oracle ASM disks on boot (y/n) [y]:
Writing Oracle ASM library driver configuration: done
```

The script prompts you for the default user and group to own the Oracle ASM driver access point. Specify the Oracle Database software owner (`grid`) and the OSDBA group (`oinstall`).

The script also prompts you to specify whether you want to start the ASMLib driver when the node is started and whether you want to scan for presence of any Oracle Automatic Storage Management disks when the node is started. Answer yes for both of these questions.

2. Repeat step 1 on each node in your cluster.

Use `ASMLib` to Create Oracle ASM Disks. Every disk that is used in an Oracle ASM disk group must be accessible on each node. After you make the physical disk available to each node, you can then mark the disk device as an Oracle ASM disk. The `/usr/sbin/oracleasm` script is used for this task.

If the target disk device supports partitioning, for example, raw devices, then you must first create a single partition that encompasses the entire disk. If the target disk device does not support partitioning, then you do not have to create a partition on the disk.

3. As the root user, use oracleasm to create Oracle ASM disks using the following syntax:

```
# fdisk /dev/sdc; fdisk /dev/sdd; fdisk /dev/sde
# /usr/sbin/oracleasm createdisk disk_name device_partition_name
```

In this command, `disk_name` is the name you choose for the Oracle ASM disk. The name you choose must contain only ASCII capital letters, numbers, or underscores, and the disk name must start with a letter, for example, `DISK1 or VOL1, or RAC_FILE1`. The name of the disk partition to mark as an Oracle ASM disk is the `device_partition_name`. For example:

```
# /usr/sbin/oracleasm createdisk DISK1 /dev/sdc1
```

If you must unmark a disk that was used in a `createdisk` command, then you can use the following syntax:

```
# /usr/sbin/oracleasm deletedisk disk_name
```

4. Repeat step 1 for each disk that is used by Oracle ASM.
5. After you have created all the Oracle ASM disks for your cluster, use the listdisks command to verify their availability:

```
# /usr/sbin/oracleasm listdisks
 DISK1
 DISK2
 DISK3
```

6. On all the other nodes in the cluster, use the scandisks command to view the newly created Oracle ASM disks. You do not have to create the Oracle ASM disks on each node, only on one node in the cluster.

```
# /usr/sbin/oracleasm scandisks
 Scanning system for ASM disks                              [OK]
```

Note: At this point, you should restart each node on which you are installing the Oracle Grid Infrastructure for cluster software. After the node has restarted, view the configured shared storage on each node. This helps to ensure that the system configuration is complete and persists across node shutdowns.

For Oracle Automatic Storage Manager (ASM) to use disks, it needs to be able to identify the devices consistently and for them to have the correct ownership and permissions. In Linux you can use ASMLib to manage these tasks, but it is seen as an additional layer of complexity and has never really gained any popularity.

By default, the Linux 2.6 kernel device file naming scheme `udev` dynamically creates device file names when the server is started, and assigns ownership of them to `root`. If `udev` applies default settings, then it changes device file names and owners for voting disks or Oracle Cluster Registry partitions, corrupting them when the server is restarted. For example, a voting disk on a device named `/dev/sdd` owned by the user `grid` may be on a device named `/dev/sdf` owned by `root` after restarting the server.

If you use ASMLib, then you do not have to ensure permissions and device path persistency in udev. If you do not use ASMLib, then you must create a custom rules file for the shared disks mounted on each node. When `udev` is started, it sequentially carries out rules (configuration directives) defined in rules files. These files are in the path `/etc/udev/rules.d/`. Rules files are read in lexical order. For example, rules in the file `10-wacom.rules` are parsed and carried out before rules in the rules file `90-ib.rules`.

Where rules files describe the same devices, on Asianux, Red Hat, and Oracle Linux, the last file read is the one that is applied. On SUSE 2.6 kernels, the first file read is the one that is applied.

In this training course, we create custom rules for shared disks:

```
[root@linux1 rules.d]# cat /etc/udev/rules.d/60-raw.rules

# Enter raw device bindings here.
#
# An example would be:
#   ACTION=="add", KERNEL=="sda", RUN+="/bin/raw /dev/raw/raw1 %N"
# to bind /dev/raw/raw1 to /dev/sda, or
#   ACTION=="add", ENV{MAJOR}=="8", ENV{MINOR}=="1", RUN+="/bin/raw /dev/raw/raw2 %M %m"
# to bind /dev/raw/raw2 to the device with major 8, minor 1.
#
# Added for ASM disk group
# to bind /dev/raw/raw1 to /dev/sdc
    ACTION=="add", KERNEL=="sdc", RUN+="/bin/raw /dev/raw/raw1 %N"
# to bind /dev/raw/raw2 to /dev/sdd
    ACTION=="add", KERNEL=="sdd", RUN+="/bin/raw /dev/raw/raw2 %N"
# to bind /dev/raw/raw3 to /dev/sde
```

```
ACTION=="add", KERNEL=="sde", RUN+="/bin/raw /dev/raw/raw3 %N"
```

To configure a rules file for disk devices, see the chapter on configuring storage in *Oracle Grid Infrastructure Installation Guide* for your platform.

Install and configure Oracle GRID

Before installing Oracle RAC or Oracle RAC One Node, you must first install the Oracle Grid Infrastructure for a cluster software, which consists of Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM), for 11*g* release 2 (11.2). After Oracle Clusterware and Oracle ASM are operational, you can use OUI to install the Oracle Database software with the Oracle RAC components. Installing Oracle RAC One Node is available starting with Oracle Database 11*g* Release 2 (11.2.0.2).

Oracle Clusterware is installed as part of the Oracle Grid Infrastructure for cluster software. OUI installs Oracle Clusterware into a directory structure that is referred to as `Grid_home`. This home is separate from the home directories of other Oracle software products installed on the same server. Because Oracle Clusterware works closely with the operating system, system administrator privileges are required for some installation tasks. In addition, some Oracle Clusterware processes must run as the special operating system user, `root`. Oracle Automatic Storage Management (Oracle ASM) is also installed in the Grid home directory.

The Oracle RAC software is installed from the Oracle Database 11*g* installation media. By default, the standard Oracle Database 11*g* software installation process installs the Oracle RAC option when OUI recognizes that you are performing the installation on a cluster. OUI installs Oracle RAC into a directory structure that is referred to as `Oracle_home`. This home is separate from the home directories of other Oracle software products installed on the same server.

Verifying My Oracle Support Credentials

When you first start OUI, you are prompted to enter your email 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 (formerly OracleMetalink) 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.
- 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. This functionality is available starting with Oracle Database 11g Release 2 (11.2.0.2)

If you choose to enable these features, then you must supply your My Oracle Support account name (your email 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 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.

Fix & Check for a Cluster Software

During installation, for certain prerequisite check failures, you can click **Fix & Check Again** to generate a fixup script (`runcluvfy.sh`). You are then prompted by OUI to run the fixup script as the root user in a separate session. You must run the script on all the nodes specified by OUI.

The Fixup script does the following:

- Checks and sets kernel parameters to values required for successful installation, including:
  - `Shared memory parameters`
  - `Semaphore parameters`
  - `Open file descriptor and UDP send/receive parameters`
- Sets permissions on the Oracle Inventory (central inventory) directory.
- Reconfigures primary and secondary group memberships for the installation owner, if necessary, for the Oracle Inventory directory, and for the operating system privileges groups.
- Sets shell limits to required values, if necessary.

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

```
$./runcluvfy.sh stage -pre crsinst -n linux1,linux2 -fixup -verbose >fixup.out
```

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.

Install Oracle grid software using Oracle Universal Installer

As the Oracle Grid Infrastructure for a cluster software owner (`grid`) user on the first node, install the Oracle Grid Infrastructure for a cluster for your cluster. Note that OUI uses Secure Shell (SSH) to copy the binary files from this node to the other nodes during the installation. OUI can configure SSH for you during installation.

Note: If you are installing Oracle Clusterware on a server that has a single-instance Oracle Database 11*g* installation, then stop the existing Oracle ASM instances, if any. After Oracle Clusterware is installed, start the Oracle ASM instances again. When you restart the single-instance Oracle database and the Oracle ASM instances, the Oracle ASM instances use the Cluster Synchronization Services Daemon (CSSD) instead of the daemon for the single-instance Oracle database.

```
# xhost +
# su - grid
$ ./runinstaller
```

Choose **Typical Installation**, and Start to Install and Configure Grid Infrastructure

Cluster Name: oracle. In the Cluster Name filed, provide a name for the cluster. The cluster name is used to identify member nodes of a cluster you create. The default value for cluster name is based on the local node name. If you change the cluster name from default, then the cluster name that you use must be globally unique throughout your enterprise.

SCAN Name: oracle-scan.localdomain. [Remember last dot '.' after com]
The Single Client Access Name (SCAN) is a hostname used to provide service access and load balance for clients to the cluster. Because the SCAN is associated with the cluster as a whole, rather than to particular node, the SCAN makes it possible to add or remove nodes from the cluster without needing to reconfigure clients. It also adds location independence for databases, so that client configuration does not have to depend on which nodes are running a particular database. Clients can continue to access the cluster in the same way as with previous releases, but Oracle recommends that clients accessing the cluster use the SCAN.

SCAN Port: The default SCAN port for the SCAN IP is port 1521, but another port can be provided if that port is not available. After installation, a TNS listener listens to this port to response to client connections to the SCAN name.

Oracle Cluster Register (OCR) files: /ocfs/storage/ocr; Store cluster and database configuration information (256 MB);

Oracle Grid Infrastructure voting disk files: /ocfs/storage/vdsk; Contain cluster membership information. If a network failure occurs, then voting disks determine cluster ownership among cluster nodes. Select voting disk locations on shared Cluster File System (CFS) partitions that have an identical path on all nodes of the cluster, and with at least 256MB of free space.

Intelligent Platform Management Interface (IPMI) specification defines a set of common interfaces to computer hardware and firmware that system administrators can use to monitor system health, and manage the server, IPMI operates independently of the operating system and allows administrators to manage a system remotely even in the absence of the operating system or the system software, or even if the monitored system is not power on. IPMI can also function when the operating system has started, and offers enhanced features when used with the system management software.

Baseboard Management Controller (BMC) is a specialized microcontroller embedded on the motherboard of many computers, especially servers. It is part of the IPMI information about the physical state of a server using sensors. The BMC manages the interface between system management software and platform hardware, and it communicates with the system administrator through a dedicated connection.

Oracle Base for Oracle Grid Infrastructure: Oracle base location is owned by the user performing the installation, you can choose a location with an existing Oracle home, or choose another directory location that does not have the structure for an Oracle base directory. Even if you do not use the same software owner to install Oracle Clusterware and ASM, and to install Oracle Database, be aware that running the root.sh script during the Oracle grid infrastructure installation changes ownership of

the home directory where Oracle Clusterware binaries are placed to **root**. For this reason, the Oracle grid infrastructure for a cluster home cannot be in the same location as other Oracle software. However, Oracle grid infrastructure for a standalone databse-Oralce Restart- can be in the same location as other Oracle software.

Oracle base location: `$ORACLE_BASE (/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: `$ORACLE_HOME (u01/app/cluster/product/11.2.0/grid)`. 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: `asmsnmp`

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 `ORACLE_SID.DB_DOMAIN`, for example `oralin`.

Cluster Registry location: For Cluster Storage Option, we can choose OCFS we defined before, vote disks and OCR files located on /ocfs/storage/, which will store cluster and database configuration information. We can also choos ASM to store oracle cluster configuration files and vote disks, a shared filesystem by Linux1 and Linux2, where to place the Oracle Cluster Registry (OCR), and which UNIX group should be given administrative privileges (OSASM) for Automatic Storage Management.

After all grid packages installed, run the scripts on Lunix1 first, then on Linux2.

```
# /u01/app/oraInventory/orainstRoot.sh
# /u01/app/cluster/product/11.2.0/grid/root.sh
```

TIPS: `De-configure crs if installation failure`

```
# /u01/app/cluster/product/11.2.0/grid/crs/install/roothas.pl -delete
```

You can find multiple IPs (oracle-scan1 [2][3], linux1-vip) on interface eth0, The Installation of Oracle Grid Infrastructure completed successfully.

```
LSNRCTL> status
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER)))
STATUS of the LISTENER
------------------------
Alias        LISTENER
Version      TNSLSNR for Linux: Version 11.2.0.1.0 - Production
Start Date   22-NOV-2014 08:23:21
Uptime       0 days 0 hr. 10 min. 54 sec
Trace Level  off
Security     ON: Local OS Authentication
SNMP         OFF
Listener Parameter File   /u01/app/cluster/product/11.2.0/grid/network/admin/listener.ora
Listener Log File         /u01/app/grid/diag/tnslsnr/linux1/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.88.121)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.88.21)(PORT=1521)))
The listener supports no services
The command completed successfully
```

$ cat listener.ora

```
LISTENER=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER))))          # line added by
Agent
LISTENER_SCAN3=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER_SCAN3))))
# line added by Agent
LISTENER_SCAN2=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER_SCAN2))))
# line added by Agent
LISTENER_SCAN1=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER_SCAN1))))
# line added by Agent
ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER_SCAN1=ON             # line added by Agent
ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER_SCAN2=ON             # line added by Agent
```

```
ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER_SCAN3=ON             # line added by Agent
ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER=ON           # line added by Agent
```

$ cat endpoints_listener.ora

```
LISTENER_LINUX1=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=linux1-vip)(PORT=1521))(ADD
RESS=(PROTOCOL=TCP)(HOST=192.168.88.121)(PORT=1521)(IP=FIRST))))          # line added by Agent
```

$ sqlnet.ora

```
# sqlnet.ora.linux1 Network Configuration File:
/u01/app/cluster/product/11.2.0/grid/network/admin/sqlnet.ora.linux1
# Generated by Oracle configuration tools.
NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)
ADR_BASE = /u01/app/grid
```

Oracle Clusterware components

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.

The Voting Disks: 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. It’s a very interesting concept like AIX volume group quorum number, any AIX volume group will be evicted (varryoff) if the available quorum number cannot meet the minimum required number.

Oracle Cluster Registry: Oracle Cluster Registry (OCR) is a file that contains information about the cluster node list and instance-to-node mapping information. 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

Oracle Clusterware Control (CRSCTL) commands

An interface between you and Oracle Clusterware, parsing and calling Oracle Clusterware APIs for Oracle Clusterware objects.

You can use CRSCTL commands to perform several operations on Oracle Clusterware, such as:

- Starting and stopping Oracle Clusterware resources
- Enabling and disabling Oracle Clusterware daemons
- Checking the health of the cluster
- Managing resources that represent third-party applications
- Integrating Intelligent Platform Management Interface (IPMI) with Oracle Clusterware to provide failure isolation support and to ensure cluster integrity
- Debugging Oracle Clusterware components

Starting 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 linux1 linux2
```

To use the previous 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
```

Stopping 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 linux1 linux2
```

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 -all -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.

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
```

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
```

Check the status of Oracle Clusterware CRS resource status check and relocation

```
# crsctl check crs
  CRS-4638: Oracle High Availability Services is online
  CRS-4537: Cluster Ready Services is online
  CRS-4529: Cluster Synchronization Services is online
  CRS-4533: Event Manager is online
```

To list the status of all Oracle Clusterware resources running on any node in the cluster, use the following command:

```
# crsctl status resource -t
  --------------------------------------------------------------------------------
  NAME         TARGET  STATE       SERVER                STATE_DETAILS
  --------------------------------------------------------------------------------
  Local Resources
  --------------------------------------------------------------------------------
  ora.LISTENER.lsnr
               ONLINE  ONLINE      linux1
               ONLINE  ONLINE      linux2
  ora.asm
               OFFLINE OFFLINE     linux1
               OFFLINE OFFLINE     linux2
  ora.eons
               ONLINE  ONLINE      linux1
               ONLINE  ONLINE      linux2
  ora.gsd
               OFFLINE OFFLINE     linux1
               OFFLINE OFFLINE     linux2
  ora.net1.network
               ONLINE  ONLINE      linux1
               ONLINE  ONLINE      linux2
```

```
ora.ons
             ONLINE  ONLINE      linux1
             ONLINE  ONLINE      linux2
ora.registry.acfs
             OFFLINE OFFLINE     linux1
             OFFLINE OFFLINE     linux2
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.LISTENER_SCAN1.lsnr
     1       ONLINE  ONLINE      linux1
ora.LISTENER_SCAN2.lsnr
     1       ONLINE  ONLINE      linux2
ora.LISTENER_SCAN3.lsnr
     1       ONLINE  ONLINE      linux2
ora.linux1.vip
     1       ONLINE  ONLINE      linux1
ora.linux2.vip
     1       ONLINE  ONLINE      linux2
ora.oc4j
     1       OFFLINE OFFLINE
ora.scan1.vip
     1       ONLINE  ONLINE      linux1
ora.scan2.vip
     1       ONLINE  ONLINE      linux2
ora.scan3.vip
     1       ONLINE  ONLINE      linux2
```

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

```
$/u01/app/cluster/product/11.2.0/grid/bin/crsctl check cluster -all
```

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

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

```
**************************************************************
Linux1:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
Linux2:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
```

```
$ crsctl relocate resource ora.scan1.vip -f
  CRS-2673: Attempting to stop 'ora.LISTENER_SCAN1.lsnr' on 'linux2'
  CRS-2677: Stop of 'ora.LISTENER_SCAN1.lsnr' on 'linux2' succeeded
  CRS-2673: Attempting to stop 'ora.scan1.vip' on 'linux2'
  CRS-2677: Stop of 'ora.scan1.vip' on 'linux2' succeeded
  CRS-2672: Attempting to start 'ora.scan1.vip' on 'linux1'
  CRS-2676: Start of 'ora.scan1.vip' on 'linux1' succeeded
  CRS-2672: Attempting to start 'ora.LISTENER_SCAN1.lsnr' on 'linux1'
  CRS-2676: Start of 'ora.LISTENER_SCAN1.lsnr' on 'linux1' succeeded

$ crsctl relocate resource ora.linux1.vip -f
  CRS-2673: Attempting to stop 'ora.linux1.vip' on 'linux2'
  CRS-2677: Stop of 'ora.linux1.vip' on 'linux2' succeeded
  CRS-2672: Attempting to start 'ora.linux1.vip' on 'linux1'
  CRS-2676: Start of 'ora.linux1.vip' on 'linux1' succeeded
  CRS-2672: Attempting to start 'ora.LISTENER.lsnr' on 'linux1'
```

```
CRS-2676: Start of 'ora.LISTENER.lsnr' on 'linux1' succeeded
```

Oracle Clusterware Control (CRSCTL) commands on Oracle entities

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.

Stop/Start CRS Resource

```
$ crsctl stop res ora.scan1.vip
```

```
$ crsctl start res ora.scan1.vip
CRS-2672: Attempting to start 'ora.scan1.vip' on 'linux1'
CRS-2676: Start of 'ora.scan1.vip' on 'linux1' succeeded
CRS-2672: Attempting to start 'ora.LISTENER_SCAN1.lsnr' on 'linux1'
CRS-2676: Start of 'ora.LISTENER_SCAN1.lsnr' on 'linux1' succeeded
```

Define/Modify/Delete CRS Resource

```
$ su root "-c crsctl modify res ora.scan1.vip -attr "attribute_name=attribute_value""
$ su root "-c crsctl modify res ora.scan1.vip -attr 'USR_ORA_VIP=192.168.189.100'"
```

```
$ crs_stat -t
Name          Type          Target    State     Host
------------------------------------------------------------
ora....ER.lsnr ora....er.type ONLINE    ONLINE    linux1
ora....N1.lsnr ora....er.type ONLINE    ONLINE    linux1
ora....N2.lsnr ora....er.type ONLINE    ONLINE    linux2
ora....N3.lsnr ora....er.type ONLINE    ONLINE    linux2
ora.asm        ora.asm.type   OFFLINE   OFFLINE
ora.eons       ora.eons.type  ONLINE    ONLINE    linux1
ora.gsd        ora.gsd.type   OFFLINE   OFFLINE
ora....SM1.asm application    OFFLINE   OFFLINE
ora....X1.lsnr application    ONLINE    ONLINE    linux1
ora.linux1.gsd application    OFFLINE   OFFLINE
ora.linux1.ons application    ONLINE    ONLINE    linux1
ora.linux1.vip ora....t1.type ONLINE    ONLINE    linux1
ora....SM2.asm application    OFFLINE   OFFLINE
ora....X2.lsnr application    ONLINE    ONLINE    linux2
ora.linux2.gsd application    OFFLINE   OFFLINE
ora.linux2.ons application    ONLINE    ONLINE    linux2
ora.linux2.vip ora....t1.type ONLINE    ONLINE    linux2
ora....network ora....rk.type ONLINE    ONLINE    linux1
ora.oc4j       ora.oc4j.type  OFFLINE   OFFLINE
ora.ons        ora.ons.type   ONLINE    ONLINE    linux1
ora....ry.acfs ora....fs.type OFFLINE   OFFLINE
ora.scan1.vip  ora....ip.type ONLINE    ONLINE    linux1
ora.scan2.vip  ora....ip.type ONLINE    ONLINE    linux2
ora.scan3.vip  ora....ip.type ONLINE    ONLINE    linux2
```

```
$ crsctl stat res ora.scan1.vip -p
NAME=ora.scan1.vip
TYPE=ora.scan_vip.type
ACL=owner:root:rwx,pgrp:root:r-x,other::r--,group:oinstall:r-x,user:grid:r-x
ACTION_FAILURE_TEMPLATE=
ACTION_SCRIPT=
ACTIVE_PLACEMENT=1
AGENT_FILENAME=%CRS_HOME%/bin/orarootagent%CRS_EXE_SUFFIX%
AUTO_START=restore
CARDINALITY=1
CHECK_INTERVAL=1
DEFAULT_TEMPLATE=PROPERTY(RESOURCE_CLASS=scan_vip)
DEGREE=1
DESCRIPTION=Oracle SCAN VIP resource
ENABLED=1
FAILOVER_DELAY=0
FAILURE_INTERVAL=0
```

```
  FAILURE_THRESHOLD=0
  HOSTING_MEMBERS=
  LOAD=1
  LOGGING_LEVEL=1
  NLS_LANG=
  NOT_RESTARTING_TEMPLATE=
  OFFLINE_CHECK_INTERVAL=0
  PLACEMENT=balanced
  PROFILE_CHANGE_TEMPLATE=
  RESTART_ATTEMPTS=0
  SCAN_NAME=oracle-scan
  SCRIPT_TIMEOUT=60
  SERVER_POOLS=*
  START_DEPENDENCIES=hard(ora.net1.network) dispersion:active(type:ora.scan_vip.type)
  pullup(ora.net1.network)
  START_TIMEOUT=0
  STATE_CHANGE_TEMPLATE=
  STOP_DEPENDENCIES=hard(ora.net1.network)
  STOP_TIMEOUT=0
  UPTIME_THRESHOLD=1h
  USR_ORA_ENV=
  USR_ORA_VIP=192.168.10.100
  VERSION=11.2.0.1.0

$ su root "-c crsctl modify res ora.scan1.vip -attr "USR_ORA_VIP=192.168.189.100""


$ crsctl add res ora.scan2.vip -type ora.scan_vip.type -attr

"SCAN_NAME=Linux-cluster,USR_ORA_VIP=192.168.189.101"

$ su root "-c crsctl setperm res ora.scan2.vip -o root"
$ su root "-c crsctl setperm res ora.scan2.vip -g root"
$ su root "-c crsctl setperm res ora.scan2.vip -u group:oinstall:r-x"
$ su root "-c crsctl setperm res ora.scan2.vip -u user:grid:r-x"
```

Administering Voting Disks for Oracle Clusterware: 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 diskgroup.

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 on stored on disk:

1. $ crsctl query css votedisk

```
 ##  STATE    File Universal Id                File Name Disk group
 --  -----    -----------------                --------- ---------
  1. ONLINE   3fb5177c06944f0abf86462580303bde (/ocfs/storage/vdsk) []
 Located 1 voting disk(s).
```

2. Run the following command as the grid (or oracle) user to add a voting disk:

```
 $ crsctl add css votedisk path
```

3. Run the following command as the grid (or oracle) user to remove a voting disk:

```
 $ crsctl delete css votedisk path
```

Backing up Voting Disks: In Oracle Clusterware 11*g* Release 2 (11.2), you no longer have to back up the voting disk. The voting disk data is automatically backed up in OCR as part of any configuration change. The voting disk files are backed up automatically by Oracle Clusterware if the contents of the files have changed in the following ways:

1. Configuration parameters, for example misscount, have been added or modified
2. After performing voting disk add or delete operations

Replace Voting Disks: If a voting disk is damaged, and no longer usable by Oracle Clusterware, then you can replace or re-create the voting disk. You replace a voting disk by deleting the unusable voting disk and then adding a new voting disk to your configuration. 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
   ```

Migrating Voting Disks to Oracle ASM Storage: Starting with Oracle Clusterware 11*g* release 2, you can 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.

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 11.2.0.0 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
```

Backing Up and Recovering the Oracle Cluster Registry: Oracle Clusterware automatically creates 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.

View Available OCR Backups: Use the ocrconfig utility to view the backups generated automatically by Oracle Clusterware. **To find the most recent backup of OCR:** Run the following command on any node in the cluster:

```
$ ocrconfig -showbackup
```

Manually Back Up OCR: Use the ocrconfig utility to force Oracle Clusterware to perform a backup of 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.

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.

(Optional) 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 your Oracle Grid Infrastructure for cluster software. 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.

Recover 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.

Check the Status of OCR: In event of a failure, before you attempt to restore the OCR, ensure that the OCR is unavailable.

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.

Restore 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.

1. Identify the available OCR backups using the ocrconfig command:

   ```
   # ocrconfig -showbackup
   ```

2. 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:
```
# ocrdump ocr_dump_output_file -backupfile file_name
```

If you do not specify an output file name, then the OCR contents are written to a file named OCRDUMPFILE in the current directory.

3. As the root user, stop Oracle Clusterware on all the nodes in your cluster by executing the following command:
```
# crsctl stop cluster -all
```

4. As the root user, restore the OCR by applying an OCR backup file that you identified in Step 1 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.
```
# ocrconfig -restore file_name
```

5. As the root user, restart Oracle Clusterware on all the nodes in your cluster by running the following command:
```
# crsctl start cluster -all
```

6. 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]
```

Change Oracle Cluster Registry Configuration

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.

Oracle Clusterware supports up to five OCR copies. You can add an OCR location after an upgrade or after completing the Oracle RAC installation. 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:
```
# ocrconfig -add new_ocr_file_name
```

This command updates the OCR configuration on all the nodes on which Oracle Clusterware is running.

Migrating OCR to Oracle ASM Storage: Starting with Oracle Clusterware 11*g* release 2, you can store the OCR in an Oracle ASM disk group. By default, the OCR is configured to use Oracle ASM when you perform a new installation of Oracle Clusterware. 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. Use the Oracle ASM Configuration Assistant (ASMCA) to create an Oracle ASM disk group that is at least the same size of the existing OCR and has at least normal redundancy.
2. Verify that the ASM Compatibility attribute for the disk group is set to 11.2.0.0 or higher.
3. Run the following OCRCONFIG command as the root user, specifying the Oracle ASM disk group name:
```
# ocrconfig -add +ASM_disk_group
```
4. 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.

Replace an OCR: If you must change the location of an existing OCR, or change the location of a failed OCR to the location of a working one, then you can use the following procedure if one OCR file remains online.

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
**************************************************************
linux1:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
linux2:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
```
3. As the root user, enter the following command to designate a new location for the specified OCR file:
```
# 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
Status of Oracle Cluster Registry is as follows :
         Version                  :          3
         Total space (kbytes)     :     262120
         Used space (kbytes)      :       2424
         Available space (kbytes) :     259696
         ID                       : 1339692072
         Device/File Name         : /ocfs/storage/ocr
                                    Device/File integrity check succeeded

                                    Device/File not configured
                                    Device/File not configured
                                    Device/File not configured
                                    Device/File not configured
         Cluster registry integrity check succeeded
         Logical corruption check bypassed due to non-privileged user
```

Remove an OCR: To remove an 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).

```
# ocrconfig -delete ocr_file_name
```

This command updates the OCR configuration on all the nodes on which Oracle Clusterware is running.

Repaire 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. 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:

```
# ocrconfig –repair -add new_ocr_file_name
# ocrconfig –repair -delete ocr_file_name
# 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.

5. Restart Oracle Clusterware on the node you have just repaired.
6. As the root user, check the OCR configuration integrity of your cluster using the following command:

```
# ocrcheck
```

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.

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.

Verifying the Integrity of Oracle Clusterware Components

You use the CVU comp crs command to verify the existence of all the 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

Verifying the Integrity of the Oracle Cluster Registry

You use the CVU comp ocr command 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

Verifying the Integrity of Your Entire Cluster

You use the CVU comp clu command to check that all nodes in the cluster have the same view of the cluster configuration.

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]
   ```

Checking 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.

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 interconnect.
-   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
```

To run the Oracle Clusterware Diagnostics Collection script

If you see that one or more Oracle Clusterware resources are offline, you can reboot your hosts (linux1, then linux2), or if they are still missing, then the Oracle Clusterware software did not install properly.

1. In a command window, log in to the operating system as the root user.
2. Run the diagcollection.pl script from the operating system prompt as follows, where Grid_home is the home directory of your Oracle Grid Infrastructure for a cluster installation:

```
# Grid_home/bin/diagcollection.pl --collect
```

To enable debugging of Oracle Clusterware components:

The Oracle Clusterware Diagnostics Collection script collects diagnostic information for your Oracle Clusterware installation. The diagnostics provide additional information so that Oracle Support Services can resolve problems. It displays the status of the Cluster Synchronization Services (CSS), Event Manager (EVM), and the Cluster Ready Services (CRS) daemons.

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 debug 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 debug log css CSSD:1
```

To enable debugging of an Oracle Clusterware resource:

1. Obtain a list of the resources available for debugging by running the following command:
   ```
   # crsctl check crs
   ```
2. 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 debug log res resource_name:debugging_level
```

Enabling Tracing

CVU does not generate trace files unless you enable tracing. The CVU trace files are created in the CRS_home/cv/log directory. Oracle RAC 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 space, if needed.

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.

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 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 *CRS_home*/log/*hostname*/alert*hostname*.log, where *CRS_home* is the directory in which Oracle Clusterware was installed and *hostname* is the host name of the local node.

About the Oracle Clusterware Component Log Files

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

In each of the following log file locations, *hostname* is the name of the node, for example, racnode2 and *CRS_home* is the directory in which the Oracle Clusterware software was installed.

The log files for the CRS daemon, crsd, can be found in the following directory:

```
CRS_home/log/hostname/crsd/
```

The log files for the CSS daemon, cssd, can be found in the following directory:

```
CRS_home/log/hostname/cssd/
```

The log files for the EVM daemon, evmd, can be found in the following directory:

```
CRS_home/log/hostname/evmd/
```

The log files for the Oracle Cluster Registry (OCR) can be found in the following directory:

`CRS_home/log/hostname/client/`

Each program that is part of the Oracle RAC high availability component has a subdirectory assigned exclusively for that program. The name of the subdirectory equals the name of the program.

Note: If any of the Oracle Clusterware components generates a core dump file, then the dump file is located in a subdirectory of the log directory for that component.

Common Oracle Cluster Registry 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.

Creating Oracle ASM Disk Groups

Automatic Storage Management (ASM) is an integrated volume manager and file system. It frees you from managing individual files and disk drivers. In order to user ASM, you need ASM instances and disk groups configured on the cluster. ASM diskgroups provide optimal shared storage for Oracle RAC databases. ASM diskgroups can also to host ASM Cluster File Systems (ACFS).

When installing ASM for a single-instance Oracle Database, DBCA creates a separate server parameter file (SPFILE) and password file for the ASM instance. When installing ASM in a clustered ASM environment where the ASM home is shared among all of the nodes, DBCA creates an SPFILE for ASM. In a clustered environment without a shared ASM home, DBCA creates a text-based initialization parameter file (PFILE) for ASM on each node.

You can use an SPFILE or PFILE as the ASM instance parameter file. If you use an SPFILE in a clustered ASM environment, then you must place the SPFILE on a shared raw device or on a cluster file system. If you do not use a shared ASM home, then the ASM instance uses a PFILE.

The same rules for file name, default location, and search order that apply to database initialization parameter files also apply to ASM initialization parameter files. For example, in single-instance UNIX and Linux Oracle Database environments, the server parameter file for ASM has the following path:

```
$ORACLE_HOME/dbs/spfile+ASM.ora
```

Oracle Database can store a data file as an Oracle ASM file in an Oracle ASM disk group, which is a collection of disks that Oracle ASM manages as a unit. Within a disk group, Oracle ASM exposes a file system interface for database files.

TIP: You can use ASM only for database and recovery files, not for your Oracle Home or for anything else. The definition of “database file” is quite broad, but does not include trace files, the alert log, the password file, or a static parameter file.

The files types that will commonly be stored as ASM files include any or all of these:

- Controlfile

- Dynamic initialization parameter file, the spfile
- Online redo log files
- Archive redo log files
- Datafiles
- Tempfiles
- RMAN backup sets
- RMAN image copies
- Flashback logs
- Controlfile autobackups
- Data Pump dump files

As this list shows, the whole database can be stored on ASM disks, as can all recoveryrelated files. You can direct the flash recovery area to an ASM disk group.

Create ASM

You can configure Oracle ASM during installation of Oracle Grid Infrastructure for a cluster. If you choose not to configure Oracle ASM during installation, then you must manually configure Oracle ASM, or perform standalone Oracle ASM disk group management.

To manually configure Oracle ASM, you must first run ASMCA to create the Oracle ASM instances and create at least one diskgroup. Once Oracle ASM is running, then you can use DBCA to create an Oracle RAC database that uses Oracle ASM for storage.

Note: If you installed the Oracle Cluster Registry (OCR) and Voting Disks on Oracle ASM as part of your Oracle Grid Infrastructure for a cluster install, then the Oracle ASM instances are created by OUI and you do not have to run ASMCA. You must use ASMCA only if you did not specify Oracle ASM storage for the OCR and Voting disks during installation.

The Cluster Synchronization Services (CSS) daemon provides cluster services for ASM, communication between the ASM and database instances, and other essential services. When DBCA creates a database, the CSS daemon is usually started and configured to start upon restart. If DBCA created the database, then you must ensure that the CSS daemon is running before you start the ASM instance.

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.

1. Start the Oracle Automatic Storage Configuration Assistant (ASMCA) from the Grid home:

```
$ /u01/app/cluster/product/11.2.0/grid/bin/asmca
```

The Oracle ASM Configuration Assistant starts, and displays the Disk Groups window.

2. Click the **Create** button at the bottom left-hand side of the window to create a disk group.

The Create Disk Group window appears.

3. Provide the following information:
   - In the **Disk Group Name** field, enter a name for the new disk group, for example, oradg.
   - Choose a Redundancy level, for example, Normal.
   - Select the disks to include in the new disk group.

   If you used ASMLIB to configure the disks for use with Oracle ASM, then the available disks are displayed if you have the Show Eligible option selected, and they have a Header Status of PROVISIONED.

4. 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.

You can find that ASM instance (+ASM1) is registered to LISTNER dynamically.

```
[grid@linux1 ~]$ lsnrctl status

LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 29-DEC-2014 07:40:40

Copyright (c) 1991, 2009, Oracle.  All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 11.2.0.1.0 - Production
Start Date                29-DEC-2014 06:56:46
Uptime                    0 days 0 hr. 43 min. 54 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /u01/app/cluster/product/11.2.0/grid/network/admin/listener.ora
Listener Log File         /u01/app/grid/diag/tnslsnr/linux1/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.88.121)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.88.21)(PORT=1521)))
Services Summary...
Service "+ASM" has 1 instance(s).
  Instance "+ASM1", status READY, has 1 handler(s) for this service...
The command completed successfully
```

Also, oracle cluster resource service osa.asm is created and started

```
[grid@linux1 ~]$ crsctl status res -t
--------------------------------------------------------------------------------
NAME          TARGET  STATE       SERVER                STATE_DETAILS
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.LISTENER.lsnr
             ONLINE  ONLINE      linux1
             ONLINE  ONLINE      linux2
ora.ORADG.dg
             ONLINE  ONLINE      linux1
             ONLINE  ONLINE      linux2
ora.asm
             ONLINE  ONLINE      linux1                Started
             ONLINE  ONLINE      linux2                Started
ora.eons
             ONLINE  ONLINE      linux1
             ONLINE  ONLINE      linux2
ora.gsd
             OFFLINE OFFLINE     linux1
             OFFLINE OFFLINE     linux2
ora.net1.network
             ONLINE  ONLINE      linux1
             ONLINE  ONLINE      linux2
ora.ons
             ONLINE  ONLINE      linux1
             ONLINE  ONLINE      linux2
ora.oradg.oradg_db1.acfs
             ONLINE  ONLINE      linux1
             ONLINE  ONLINE      linux2
ora.oradg.oralv.acfs
             ONLINE  OFFLINE     linux1
             ONLINE  ONLINE      linux2
ora.registry.acfs
             ONLINE  ONLINE      linux1
             ONLINE  ONLINE      linux2
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.LISTENER_SCAN1.lsnr
     1       ONLINE  ONLINE      linux2
ora.LISTENER_SCAN2.lsnr
```

```
     1       ONLINE  ONLINE       linux1
ora.LISTENER_SCAN3.lsnr
     1       ONLINE  ONLINE       linux1
ora.linux1.vip
     1       ONLINE  ONLINE       linux1
ora.linux2.vip
     1       ONLINE  ONLINE       linux2
ora.oc4j
     1       OFFLINE OFFLINE
ora.scan1.vip
     1       ONLINE  ONLINE       linux2
ora.scan2.vip
     1       ONLINE  ONLINE       linux1
ora.scan3.vip
     1       ONLINE  ONLINE       linux1
```

After creating diskgroup, it's very straight forward to create volumes in diskgroups, and create ACFS file systems on volumes.

ASM Dynamic Volume Manager (ADVM) driver must be loaded on all nodes of cluster

Create ASM Volume and Create Database Home File System of ASM Cluster File System; ACFS would be registered with Grid Infrastructure as a managed resource. Grid Infrastructure uses this resource to maintain database, ASM and ACFS resource dependencies

Also, ASM configuration operations must be performed using Automatic Storage Management Configuration Assistant (asmca) from ORACLE Grid Infrastructure home (ORACLE_SID=+ASM1)

Manage ASM

To start the instance, you must connect to it as SYSASM and issue a STARTUP command. The connection can be made by setting the ORACLE_SID environment variable to the instance name (not forgetting that it must be prefixed with a “+” symbol), or if a password file has been created and enabled as in the preceding example, you can connect with password file authentication. The startup will first go through NOMOUNT, where the instance is built in memory and the disks identified by the ASM_DISKSTRING parameter are discovered. Then the instance will mount the disk groups specified by ASM_DISKGROUPS. There is no MOUNT or OPEN mode for an ASM instance; disk groups can be mounted or unmounted

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.See "Oracle Instances and Oracle ASM Storage" for an illustration of the relationship between Oracle RAC instances, Oracle ASM instances, and the shared storage. 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.

```
[root@Linux1 ~]# su - grid
[grid@Linux1 ~]$ export ORACLE_SID=+ASM1
[grid@Linux1 ~]$ sqlplus /nolog
    SQL*Plus: Release 11.2.0.1.0 Production on Thu Apr 10 15:24:43 2014

    Copyright (c) 1982, 2009, Oracle.  All rights reserved.

    SQL> conn sys as sysasm
    Enter password: sys
    Connected.
```

Drop diskgroup:

```
    SQL> alter diskgroup oradg dismount force;
    Diskgroup altered.
    SQL> drop diskgroup oradg force including contents;
    Diskgroup dropped.
```

+ASM not registers to LISTENER automatically:

```
    SQL> show parameter local_listener

    NAME                                 TYPE        VALUE
    ------------------------------------ ----------- ------------------------------
    local_listener                       string      (DESCRIPTION=(ADDRESS_LIST=(AD
                                                     DRESS=(PROTOCOL=TCP)(HOST=Linu
                                                     x1-vip)(PORT=51521))))
    SQL> alter system set local_listener =
    "(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=Linux1-vip)(PORT=1521))))";

    System altered.
```

Resize diskgroup after disk size (LUN size) changed (expanded)

```
# fdisk/dev/sdc
Delete old partition 1, and create new partition 1 with full disk cylindas

SQL> alter diskgroup oradg resize all size 9g;
     alter diskgroup oradg resize all size 9g
     *
     ERROR at line 1:
     ORA-15032: not all alterations performed
     ORA-15289: ASM disk ORADG_0002 cannot be resized beyond 3067 M
```

```
    ORA-15289: ASM disk ORADG_0001 cannot be resized beyond 3067 M
    ORA-15289: ASM disk ORADG_0000 cannot be resized beyond 3067 M


SQL> alter diskgroup oradg resize all size 9201m;
    alter diskgroup oradg resize all size 9201m
    *
    ERROR at line 1:
    ORA-15032: not all alterations performed
    ORA-15289: ASM disk ORADG_0002 cannot be resized beyond 3067 M
    ORA-15289: ASM disk ORADG_0001 cannot be resized beyond 3067 M
    ORA-15289: ASM disk ORADG_0000 cannot be resized beyond 3067 M


SQL> alter diskgroup oradg
      2  resize oradg_0000
      3  size 3067m;
    resize oradg_0000
      *
ERROR at line 2:
ORA-00905: missing keyword


SQL> alter diskgroup oradg
      2  resize disk oradg_0000
      3  size 3067m;

Diskgroup altered.

SQL> alter diskgroup oradg
      2  resize disk oradg_0001
      3  size 3067m;

Diskgroup altered.
```

Shutdown ASM instance

```
SQL> shutdown normal
  ASM diskgroups volume disabled
  ASM diskgroups dismounted
  ASM instance shutdown

SQL> quit
```

In order to make administration of ASM files simpler, Oracle provides a command-line utility that gives a Unix-like interface to ASM. This is the `ASMCMD` utility, `named %ORACLE_HOME%\bin\asmcmd.bat` on Windows and `$ORACLE_HOME/bin/asmcmd` on UNIX. To run the tool, you must first set your `ORACLE_SID` environment variable to the name of the ASM instance, and then a set of commands are available with functionality and syntax similar to that of a Unix file system

```
ASMCMD> help
  asmcmd [-vV] [-a <sysasm|sysdba>] [-p] [command]

  The environment variables ORACLE_HOME and ORACLE_SID determine the
  instance to which the program connects, and ASMCMD establishes a
  bequeath connection to it, in the same manner as a SQLPLUS / AS
  SYSASM.  The user must be a member of the OSASM group.

  Specifying the -V option prints the asmcmd version number and
  exits immediately.

  Specifying the -v option prints extra information that can help
  advanced users diagnose problems.

  Specify the -a option to choose the type of connection. There are
  only two possibilities: connecting as SYSASM or as SYSDBA.
  The default value if this option is unspecified is SYSASM.

  Specifying the -p option allows the current directory to be displayed
```

```
in the command prompt, like so:

ASMCMD [+DATA/ORCL/CONTROLFILE] >

[command] specifies one of the following commands, along with its
parameters.

Type "help [command]" to get help on a specific ASMCMD command.

commands:
      --------

      md_backup, md_restore

      lsattr, setattr

      cd, cp, du, find, help, ls, lsct, lsdg, lsof, mkalias
      mkdir, pwd, rm, rmalias

      chdg, chkdg, dropdg, iostat, lsdsk, lsod, mkdg, mount
      offline, online, rebal, remap, umount

      dsget, dsset, lsop, shutdown, spbackup, spcopy, spget
      spmove, spset, startup

      chtmpl, lstmpl, mktmpl, rmtmpl

      chgrp, chmod, chown, groups, grpmod, lsgrp, lspwusr, lsusr
      mkgrp, mkusr, orapwusr, passwd, rmgrp, rmusr

      volcreate, voldelete, voldisable, volenable, volinfo
      volresize, volset, volstat

ASMCMD> ls -l
State   Type   Rebal  Name
MOUNTED  EXTERN  N     ASMDG/
ASMCMD> cd asmdg
ASMCMD> mkdir orarac
ASMCMD> cd orarac
ASMCMD> mkdir oradata
ASMCMD> mkdir fra
ASMCMD> ls -l
Type  Redund  Striped  Time           Sys  Name
                                  N   controlfile/
                                  N   fra/
                                  N   oradata/
```

You can also perform administrative operations on Oracle ASM disk groups using Enterprise Manager such as adding and deleting them. You can also monitor Oracle ASM disk group performance and control disk group availability at the instance level. For example, Enterprise Manager provides some Oracle RAC-specific features for Oracle ASM, such as the following:

- When you add a disk group, the disk group definition includes a check box to indicate whether the disk group is automatically mounted to all the cluster database instances.
- The default Disk Group Performance page displays instance-level performance details when you click a performance characteristic such as **Write Response Time** or **I/O Throughput**.
- When you mount and dismount Oracle ASM disk groups, you can use a check box to indicate which instances should mount or dismount a particular Oracle ASM Disk Group.

In all these examples, you do not specify a filename, only a disk group. ASM will generate the actual filenames according to its own conventions. If you wish, you can see the names by querying views such as V$DATAFILE, V$LOGFILE, or V$ARCHIVED_LOG, but there is little value in this. A feature of ASM is that it gives you complete independence from the physical storage: there is no reason for you to want to know

the actual filenames. It is possible to interpret the system-generated names, but they are not any sort of physical path.

Add disk space

On both Linux1/2

Type the following to send a rescan request:

```
# echo "- - -" > /sys/class/scsi_host/host0/scan
# fdisk -l
```

You will find the new added disk, and partition this new disk (reboot if you cannot find new disk)

```
# fdisk /dev/sdf
```

```
linux1# /usr/sbin/oracleasm createdisk DISK3 /dev/sdf1
linux1# /usr/sbin/oracleasm listdisks
DISK1
DISK2
DISK3
DISK4
linux2# /usr/sbin/oracleasm scandisks
```

Oracle Instance and Database

As we discussed before, all data processing occurs in memory, in the instance, but data storage occurs in the database on disk. The database consists of three file types: the controlfile, the online redo log files, and the datafiles. Data is stored in the datafiles.

You need to understand relationship of Oracle Instance and Databas. An instance is associated with one and only one database (which is different with other RDBMS like DB2 and Informix, one instance can have more then one database). In an Oracle Real Application Clusters configuration, multiple instances access a single database simultaneously.

Every running Oracle database is associated with at least one Oracle database instance. Because an instance exists in memory and a database exists on disk, an instance can exist without a database and a database can exist without an instance, a running instance can exit without a database, but a running database cannot exist without a running instance.

A database name is uniquely identified by Global Database Name; it's also referenced by an Oracle instance on each cluster database node. Specify a prefix to be used to name the cluster database instance. The system identifier (SID) is a unique name for an Oracle database instance on a specific host. On UNIX and Linux, Oracle Database uses the SID and Oracle home values ($ORACLE_HOME) to create a key to shared memory. Also, the SID is used by default to locate the parameter file, which is used to locate relevant files such as the database control files.

In principle, when a database server crash, everything in server memory is lost, including database buffer cache, where your changes take place, and may or may not be writen to disk; instance recovery is nothing more than using the contents of the online log files to rebuild the database buffer cache to the status it was in before the crash. This rebuilding process replays all changes extracted from the redo logs that refer to blocks that had not been written to data files at the time of the crash. Once this has been done, the database can be opened. At that point, the database is still corrupted - but there is no reason not to allow users to connect, because the instance (which is what users see) has been repaired. This phase of recovery, known as the roll forward, reinstates all changes - changes to data blocks and changes to undo blocks - for both committed and uncommitted transactions.

Each redo record has the bare minimum of information needed to reconstruct a change: the block address and the new values. During roll forward, each redo record is read, the appropriate block is loaded from the datafiles into the database buffer cache, and the change is applied. Then the block is written back to data files (to disk). Once the roll forward is complete, it is as though the crash (loss of changes in memory) had never occurred.

But at that point, some users may not commit what they did yet, there will be uncommitted transactions in the database - these must be rolled back, and Oracle will do that automatically in the rollback phase of instance recovery. However, that happens after the database has been opened for use. If a user connects and hits some data that needs to be rolled back and hasn't yet been, this is not a problem - the roll forward phase will have also populated the undo segment (Not using online redo logs this time) that was protecting the uncommitted transaction, so the server can roll back the change in the normal manner for read consistency.

Because LGWR flushes all changes to all blocks to the log files, and LGWR always writes ahead of DBWn (responsible for checkpoints), and because LGWR writes in real time on commit, there will always be enough information in the redo stream to reconstruct any committed changes that had not been written to the datafiles, and to roll back any uncommitted changes that had been written to the data files. This instance recovery mechanism of redo and rollback makes it absolutely impossible to corrupt an Oracle database - so long as there has been no physical damage.

Instance recovery is automatic and unavoidable by issuing a STARTUP command. On starting an instance, the description of how SMON opens a database. First, it reads the controlfile when the database transitions to mount mode. Then in the transition to open mode, SMON checks the file headers of all the datafiles and online redo log files. At this point, if there had been an instance failure, it is apparent because the file headers are all out of sync. So SMON goes into the instance recovery routine, and the database is only actually opened after the roll forward phase has completed.

The parameter which defines and tunes instance recovery time is FAST_START_MTTR_TARGET, by defaults to zero. Zero FAST_START_MTTR_TARGET has the effect of maximizing performance, with the possible cost of long instance recovery times after an instance failure. The DBWn process will write as little as it can get away with, meaning that the checkpoint position (the point in the redo stream from which instance recovery must start following a crash) is advanced automatically by the DBWn. may be a long way out of date and that therefore a large amount of redo would have to be applied to the datafiles in the roll forward phase of instance recovery.

When and only when the database is shutdown with the NORMAL, IMMEDIATE, or TRANSACTIONAL option, there is a full checkpoint: all dirty buffers are flushed to disk by the DBWn before the database is closed and dismounted. This means that when the database is opened again, no recovery will be needed. A clean shutdown is always desirable and is necessary before some operations (such as enabling the archivelog mode or the flashback database capability). A full checkpoint can be signaled at any time with this command: alter system checkpoint; in normal running, there might be several hundred thousand dirty buffers, but the DBWn would write just a few hundred of them for the incremental checkpoint: partial checkpoints.

Install the Oracle Database Software

Install and Create Oracle database on windows server

ORACLE_HOSTNAME=orawin, ORACLE_SID=orawin. We will use this database to restore/recovery production RAC database on linux servers.

You should have a DNS server to resolve this host name (`orawin`) to an IP (`192.168.88.1`) or use local hosts to resolve, the local hosts file is on: `%SystemRoot%\system32\drivers\etc\hosts`.

In previous releases of Oracle Database, you were required to set environment variables for `ORACLE_HOME` and `ORACLE_SID` to start, stop, and check the status of Enterprise Manager. With Oracle Database 11g release 2 (11.2) and later, you need to set the environment variables `ORACLE_HOME` and `ORACLE_UNQNAME` to use Enterprise Manager.

Oracle software and database installation information using OUI tools on Windows:

Using “`regedit`”, get Oracle registry keys on Windows OS (equalvent to environment variables on Liunx/Unix):

During development, even in normal operation, user errors were undoubtedly the worst errors to manage. Recent releases of the database improve the situation dramatically. Oracle provides a number of ways whereby you as DBA may be able to correct user errors, but this is often extremely difficult - particularly if the error isn’t reported for some time. The possible techniques include `flashback query, flashback drop, and flashback database` and `incomplete recovery.`
`Flashback query` involves running a query against a version of the database as at some time in the past. The read-consistent version of the database is constructed, for your session only, through the use of undo data.

`Flashback drop` reverses the effect of a `DROP TABLE` command. In previous releases of the database, a DROP command did what it says: it dropped all references to the table from the data dictionary. There was no way to reverse this. Even flashback query would fail, because the flashback query mechanism does need the data dictionary object definition. But from release 10g the implementation of the DROP command has changed: it no longer drops anything; it just renames the object so that you will never see it again, unless you specifically ask to.

Operating System Environment for Oracle Grid Infrastructure on Linux

Now that the Oracle Grid Infrastructure for cluster software is functional, you can install the Oracle Database software on the ones of your cluster.

Before you install Oracle RAC, you should make sure Virtual IP (VIP) run on its own server, otherwise, RAC installation will fail.

You run OUI from the `oracle` user account. Before you start OUI you do not have to configure the environment of the `oracle` user.

However, you can set the `ORACLE_BASE` environment variable to the directory in which you want the Oracle Inventory files located.For example, if you plan to make the Oracle Database home directory `/u01/app/oracle/product/11.2.0/dbhome_1`, then you would set `ORACLE_BASE` to the directory `/u01/app/oracle`. If you set the `ORACLE_BASE` directory before installation, then this 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` directory before installation, then this becomes the default location for the Grid home displayed by OUI.

```
# su - oracle
$ cd /u01/app/cluster/product/11.2.0/grid/bin
$ ./cluvfy stage -pre dbinst -fixup -n linux1,linux2 -osdba dba -verbose
```

Install Oracle RAC Software using Oracle Universal Installer

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. If you plan to use Oracle RAC One Node, then you must first install the Oracle RAC software without creating a database. After the installation completes, you use Database Configuration Assistant (DBCA) to create the Oracle RAC One Node database.

To launch the DBCA on Linux, first set the environment variables that should always be set for any Linux DBA session: ORACLE_BASE, ORACLE_HOME, PATH, and LD_LIBRARY_PATH.

Add following in oracle user's .bash_profile:

```
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export PATH=$PATH:$HOME/bin:$ORACLE_HOME/bin:/bin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/X11R6/bin
```

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:

   ```
   ./runInstaller
   ```

   When you start Oracle Universal Installer, the Configure Security Updates window appears.

2. (Optional) Enter your `email address` and My Oracle Support `password`, and 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.

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

3. If you want to create an Oracle RAC database, then select **Create and configure a database**. If you want to create an Oracle RAC One Node database, then select **Install database software only**. Click **Next** to continue.

The System Class window appears.

4. Select the **Real Application Clusters database** installation type.

   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.

   Click the **SSH Connectivity** button at the bottom of the window. The bottom panel of the window displays the SSH Connectivity information

5. 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**. If you are using a network user with home directory on shared storage, then also select the "User home if shared by the selected nodes" option. Click **Test**.

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

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

   The Select Install Type window appears.

7. 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.

8. 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 chose, 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.

  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 `ORACLE_SID.DB_DOMAIN`, for example, oralin.linux.lii01.livun.com.

  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 eight 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. After you have provided all the necessary information, click **Next**. The Perform Prerequisite Checks window appears.

9. After a short period, the Summary window appears. Review the information on this screen, 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.
10. 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, including the URL for the Database Control console. 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 this 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.

11. 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.

    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):
    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.
```

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

The Install Product window is displayed.

12. Click **Close** to exit the installer.
13.

Server Parameter File for Oracle Real Application Clusters

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. In the environment described by this guide, the SPFILE is created on an Oracle ASM disk group.

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 2 Day DBA*
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about using a server parameter file in an Oracle Real Application Clusters environment
- *Oracle Database Backup and Recovery User's Guide* for information about using RMAN to backup the SPFILE

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.

The administration of initialization parameters in an Oracle RAC environment is slightly different from single-instance database parameter administration. For example, if you change a parameter setting that is marked by an asterisk, which indicates that the parameter is a clusterwide 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
- Editing Initialization Parameter Settings for an Oracle RAC Database
- About the Server Parameter File for Oracle Real Application Clusters

See Also:

- *Oracle Database 2 Day DBA*
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about initialization parameters in an Oracle RAC environment

Configuring Initialization Parameters for an Oracle RAC Database

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 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:

- Parameters that Must Have Identical Settings on All Instances
- Parameters that Must Have Unique Settings on All Instances
- Parameters that Should Have Identical Settings on All Instances

Parameters that Must Have Identical Settings on All Instances

Certain initialization parameters that are critical at database creation or that affect certain database operations must have the same value for every instance in an Oracle RAC database. Specify these parameter values in the SPFILE, or within the individual PFILEs for each instance. The following list contains the parameters that must be identical on every instance:

- `CLUSTER_DATABASE`
- `COMPATIBLE`
- `CONTROL_FILES`
- `DB_BLOCK_SIZE`
- `DB_DOMAIN`
- `DB_FILES`
- `DB_NAME`
- `DB_RECOVERY_FILE_DEST`

- DB_RECOVERY_FILE_DEST_SIZE
- DB_UNIQUE_NAME
- DML_LOCKS (only if set to 0)
- INSTANCE_TYPE (RDBMS or ASM)
- PARALLEL_EXECUTION_MESSAGE_SIZE
- REMOTE_LOGIN_PASSWORDFILE
- RESULT_CACHE_MAX_SIZE (either enabled or disabled on all instances)
- UNDO_MANAGEMENT

Parameters that Must Have Unique Settings on All Instances

For the following parameters, the parameter must be set on all instances to a value that is unique to that instance:

- CLUSTER_INTERCONNECTS
- INSTANCE_NUMBER
- ROLLBACK_SEGMENTS (if the UNDO_MANAGEMENT initialization parameter is not set to AUTO)
- UNDO_TABLESPACE (if UNDO_MANAGEMENT is set to AUTO)

Oracle Database uses the INSTANCE_NUMBER parameter to distinguish among instances at startup. Oracle Database uses the INSTANCE_NAME parameter to assign redo log groups to specific instances. To simplify administration, use the same number for both the INSTANCE_NAME and INSTANCE_NUMBER parameters. With Oracle Database 11.2 using Grid Plug and Play, the instance name parameter is no longer required. If you do not specify the instance name, then instance name defaults to *db_unique_name_number*.

Parameters that Should Have Identical Settings on All Instances

Oracle recommends that you set the values for the following parameters to the same value on all instances. Although you can have different settings for these parameters on different instances, setting each parameter to the same value on all instances simplifies administration:

Parameter	**Description**
ARCHIVE_LAG_TARGET	Different values for instances in your Oracle RAC database are likely to increase overhead because of additional automatic synchronization performed by the database processing. When using Oracle Streams with your Oracle RAC database, the value should be greater than zero.
CONTROL_MANAGEMENT_PACK_ACCESS	This parameter controls the use of the Diagnostics and Tuning Packs for Oracle Enterprise Manager Database Control (Database Control). You should set the value for this parameter on all instances to reflect whether you have purchased the Diagnostics and Tuning Manageability Packs for your Oracle RAC database.
DIAGNOSTIC_DEST	As of Oracle Database 11*g* release 1 (11.1), the diagnostics for each database instance are located in a dedicated directory, which can be specified through the DIAGNOSTIC_DEST initialization parameter. This parameter can be set on each instance. Oracle recommends that each instance in a cluster specify a

Parameter	Description
	DIAGNOSTIC_DEST directory location that is located on shared disk and that the same value for DIAGNOSTIC_DEST is specified for each instance.
LICENSE_MAX_USERS	This parameter determines a database-wide limit on the number of user accounts defined in the database and it is useful to have the same value on all instances of your database so you can see the current value no matter which instance you are using. Setting different values may generate additional warning messages during instance startup, or cause commands related to database user account management to fail on some instances. If you have concurrent usage (session) licensing, then set the LICENSE_MAX_SESSIONS parameter instead. When using LICENSE_MAX_SESSIONS, set the value on each instance so that the sum of all the values is less than or equal to the total number of sessions licensed for the database. Do not set both LICENSE_MAX_SESSIONS and LICENSE_MAX_USERS.
LOG_ARCHIVE_FORMAT	If you do not use the same value for all your instances, then you complicate media recovery. The recovering instance expects the required archived redo log file names to have the format defined by its own value of LOG_ARCHIVE_FORMAT, regardless of which instance created the archived redo log files. Databases that support Oracle Data Guard, either to send or receive archived redo log files, must use the same value of LOG_ARCHIVE_FORMAT for all instances.
REDO_TRANSPORT_USER	This parameter specifies the name of the user whose password verifier is used when a remote login password file is used for redo transport authentication. This parameter is used in Oracle Data Guard configurations.
SPFILE	The SPFILE resides in the $ORACLE_HOME/dbs directory; however, users can place it any storage accessible to the local host if it is specified in an initialization parameter file. If this parameter does not identify the same file to all instances, then each instance may act differently and unpredictably in failover, load-balancing, or standard operations. Additionally, a change you make to the SPFILE using an ALTER SYSTEM SET or ALTER SYSTEM RESET command is saved only in the SPFILE used by the instance where you run the command. Your change is not reflected in instances using different SPFILEs.
TRACE_ENABLED	When TRACE_ENABLED is set to true, Oracle records information in specific files when errors occur. Oracle Support Services uses this information for debugging the Oracle software. If you enable tracing on one instance, then you should enable it on all instances to ensure that the diagnostic information is recorded in a trace file.
UNDO_RETENTION	By setting different values for UNDO_RETENTION in each instance, you are likely to reduce scalability and encounter unpredictable actions following a failover. Therefore, you should carefully consider the potential benefits before you assign

Parameter	Description
	different values for this parameter to the instances in your Oracle RAC database.

SERVICE_NAMES Parameter for Oracle RAC

Do not set the SERVICE_NAMES parameter for Oracle RAC environments. Instead, define services using Oracle Enterprise Manager and manage those services using Server Control (SRVCTL) utility.

The `SERVICE_NAMES` initialization parameter specifies one or more names by which clients can connect to the instance (Default value DB_UNIQUE_NAME.DB_DOMAIN if defined). 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.

Database services (services) are logical abstractions for managing workloads in Oracle Database. Services divide workloads into mutually disjoint groupings. Each service represents a workload with common attributes, service-level thresholds, and priorities. The grouping is based on attributes of work that might include the application function to be used, the priority of execution for the application function, the job class to be managed, or the data range used in the application function or job class. For example, the Oracle E-Business suite defines a service for each responsibility, such as general ledger, accounts receivable, order entry, and so on. Each database service has a unique name.

Connection requests can include a database service name. If no service name is included and the Net Services file listener.ora designates a default service, the connection uses the default service.

Services are built into the Oracle Database, providing a single system image for workloads, prioritization for workloads, performance measures for real transactions, and alerts and actions when performance goals are violated. Services enable you to configure a workload, administer it, enable and disable it, and measure the workload as a single entity. You can do this using standard tools such as the Database Configuration Assistant (DBCA), Net Configuration Assistant (NetCA), and Enterprise Manager (EM). Enterprise Manager supports viewing and operating services as a whole, with drill down to the instance-level when needed.

In Real Application Clusters (RAC), a service can span one or more instances and facilitate real workload balancing based on real transaction performance. This provides end-to-end unattended recovery, rolling changes by workload, and full location transparency. RAC also enables you to manage a number of service features with Enterprise Manager, the DBCA, and the Server Control utility (SRVCTL).

Services also offer an extra dimension in performance tuning. Tuning by "service and SQL" can replace tuning by "session and SQL" in the majority of systems where all sessions are anonymous and shared. With services, workloads are visible and measurable. Resource consumption and waits are attributable by application. Additionally, resources assigned to services can be augmented when loads increase or decrease. This dynamic resource allocation enables a cost-effective solution for meeting demands as they occur. For example, services are measured automatically and the performance is compared to service-level thresholds. Performance violations are reported to Enterprise Manager, enabling the execution of automatic or scheduled solutions.

Deploying Services:

When you configure database services, you give each service a unique global name, associated performance goals, and associated importance. The services are tightly integrated with Oracle Database and are maintained in the data dictionary. You can find service information in the following service-specific views:

- DBA_SERVICES
- ALL_SERVICES or V$SERVICES
- V$ACTIVE_SERVICES
- V$SERVICE_STATS
- V$SERVICE_EVENTS
- V$SERVICE_WAIT_CLASSES
- V$SERV_MOD_ACT_STATS
- V$SERVICE_METRICS
- V$SERVICE_METRICS_HISTORY

The following additional views also contain some information about services:

- V$SESSION
- V$ACTIVE_SESSION_HISTORY
- DBA_RSRC_GROUP_MAPPINGS
- DBA_SCHEDULER_JOB_CLASSES
- DBA_THRESHOLDS

Several Oracle Database features support services. The Automatic Workload Repository (AWR) manages the performance of services. AWR records service performance, including execution times, wait classes, and resources consumed by service. AWR alerts warn when service response time thresholds are exceeded. The dynamic views report current service performance metrics with one hour of history. Each service has quality-of-service thresholds for response time and CPU consumption.

In addition, the Database Resource Manager maps services to consumer groups. This enables you to automatically manage the priority of one service relative to others. You can use consumer groups to define relative priority in terms of either ratios or resource consumption. This is described in more detail, for example, in Oracle Real Application Clusters Deployment and Performance Guide.

Configuring Services

Services describe applications, application functions, and data ranges as either functional services or data-dependent services. Functional services are the most common mapping of workloads. Sessions using a particular function are grouped together. For Oracle*Applications, ERP, CRM, and iSupport functions create a functional division of the work. For SAP, dialog and update functions create a functional division of the work.

In contrast, data-dependent routing routes sessions to services based on data keys. The mapping of work requests to services occurs in the object relational mapping layer for application servers and TP monitors. For example, in RAC, these ranges can be completely dynamic and based on demand because the database is shared.

You can also define preconnect application services in RAC databases. Preconnect services span instances to support a service in the event of a failure. The preconnect service supports TAF preconnect mode and is managed transparently when using RAC.

In addition to application services, Oracle Database also supports two internal services: SYS$BACKGROUND is used by the background processes only and SYS$USERS is the default service for user sessions that are not associated with services.

Use the DBMS_SERVICE package or set the SERVICE_NAMES parameter to create application services on a single-instance Oracle Database. You can later define the response time goal or importance of each service through EM, either individually or by using the Enterprise Manager feature "Copy Thresholds From a Baseline" on the Manage Metrics/Edit Threshold pages. You can also do this using PL/SQL.

Using Services

Using services requires no changes to your application code. Client-side work connects to a service. Server-side work specifies the service when creating the job class for the Job Scheduler and the database links for distributed databases. Work requests executing under a service inherit the performance thresholds for the service and are measured as part of the service.

Client-Side Use

Middle-tier applications and client-server applications use a service by specifying the service as part of the connection in TNS connect data. This connect data may be in the TNSnames file for thick Net drivers, in the URL specification for thin drivers, or may be maintained in the Oracle Internet Directory. For example, data sources for the Oracle Application Server are set to route to a service. Using easy connect naming, this connection needs only the host name and service name (for example, hr@myDBhost/myservice). For Oracle E-Business Suite, the service is also maintained in the application database identifier and in the cookie for the ICX parameters.

Server-Side Use

Server-side work, such as Oracle Scheduler, parallel execution, and Oracle Streams Advanced Queuing, set the service name as part of the workload definition.

For Oracle Scheduler, the service that the job class uses is optionally defined when the job class is created. During execution, jobs are assigned to job classes, and job classes can run within services. Using services with job classes ensures that the work executed by the job scheduler is identified for workload management and performance tuning.

For parallel query and parallel DML, the query coordinator connects to a service just like any other client. The parallel query processes inherit the service for the duration of the execution. At the end of query execution, the parallel execution processes revert to the default service.

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.

See Also:

- "About Oracle Services"

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.

To view or modify the initialization parameters using Enterprise Manager:

1. On the Cluster Database Home page, while logged in as a SYSDBA user, click **Server**.

   The Server page appears.

2. Under the heading Database Configuration, select **Initialization Parameters**.

   The Initialization Parameters page appears.

3. Select either the **Current** or **SPFile** subpage to modify the parameter settings.

   See either:
   - "Modifying the Initialization Parameter for Oracle RAC Using the Current Tab"
   - "Modifying the Initialization Parameter for Oracle RAC Using the SPFile Tab"

Modifying the Initialization Parameter for Oracle RAC Using the Current Tab

The Current subpage of the Initialization Parameters page contains a list of configuration parameters for that instance and database. You can set these parameters 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.

The Instance column shows the instances for which the parameter has the value listed in the table. An asterisk (*) indicates that the parameter has the same value for all remaining instances of the cluster database. For example, if `open_cursors = 250` for `node1` and `open_cursors = 300` for `node2`, then the Instance column for `open_cursors = 250` displays an asterisk, but the Instance column for `open_cursors = 300` contains "node2". This shorthand notation saves space when the cluster databases has many instances.

Description of the illustration init_params1.gif

You can filter the Initialization Parameters page to show only those parameters that meet the criteria of the filter you enter in the Filter by name field. Optionally, you can select **Show All** to display on one page all parameters currently used by the running instance(s).

To modify parameter values using the Current tab:

1. Select a parameter from the Select column and do one of the following:
   - Click **Add** to add the selected parameter to a different instance.

     When the new row appears in the table, choose the instance name to which the new setting applies, and enter the new value for the parameter in the Value field. Optionally, you can put text in the Comment field to indicate why the instance has a different value from the other instances.

   - Click **Reset** to reset the value of the selected parameter. Note that using this option you can reset only those parameters that do not have an asterisk in the Instance column.

     When you click Reset, the value of the selected parameter is reset to the value of the remaining instances (that is, the row with the asterisk in the Instance column) and the row that has the local instance name in the Instance field is removed from the table.

2. For example, select the parameter OPEN_CURSORS, then click Add. In the new entry for OPEN_CURSORS, select an Instance (for example sales1 or RACDB2), change the Value field to 300.
3. After you make changes to one or more of the parameters, click Apply to accept and implement the changes.

Modifying the Initialization Parameter for Oracle RAC Using the SPFile Tab

You can also add or reset parameters using the SPFile tab. When you modify initialization parameters using the SPFile tab, 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 tab is different than resetting the same parameters using the Current tab. 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.

- If you reset a parameter with an asterisk in the Instance column, then the entry is deleted from both the SPFILE and the table in Database Control. Only parameters without an asterisk (or instance-specific parameters) remain.

Description of the illustration init_params2.gif

- If you reset the only entry for a nonasterisk parameter, then it is deleted from both the SPFILE and the table in Database Control, but the parameter is replaced by a *dummy* parameter with an empty value field and an asterisk in the Instance column. This enables you to specify a new value for the parameter, add new instance-specific entries for the parameter, and so on.

Description of the illustration init_params3.gif

Resetting a parameter that is set for only one instance results in the parameter being unset for all instances.

To modify parameter values using the SPFile tab:

1. Choose a parameter using the **Select** column.

   For more information about each parameter, click the information icon in the Help column next to the parameter.

2. 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.
3. Click **Apply** to apply the changes to the SPFILE, or to the SPFile and the currently running instance if you selected the "Apply changes in SPFile mode to the current running instance(s)" option.
4. On the SPFile subpage of the Initialization Parameters page, select a parameter from the Select column and do one of the following:
   - Click **Add** to add the selected parameter to a different instance.

When the new row appears in the table, choose the instance name to which the new setting applies, and then enter the new value for the parameter in the Value field. Optionally, you can put text in the Comment field describing the reason for the change.

- Click **Reset** to reset the value of the selected parameter.

  When you click Reset, one of the following actions is performed:

  - If the entry you selected was for a specific instance, then the value of the selected parameter for that instance is reset to the value of the remaining instances (indicated by an asterisk in the Instance column). The entry that has the local instance name in the Instance field is deleted.
  - If the entry you selected to reset was the default values for all instances (indicated by an asterisk in the Instance column), then the value of the selected parameter is unset for all instances, but any instance-specific parameter entries for the same parameter are not changed.
  - If you reset the only entry for a parameter, regardless of whether the entry applies to all instances or a single instance, then the parameter is unset for all instances in the cluster database.

5. After you make changes to one or more of the parameters, click **Apply** to accept and implement the changes.

Example: Modifying the OPEN_CURSORS Parameter

Suppose that the `open_cursors` parameter has two entries in the SPFILE:

```
*.open_cursors = 200
RACDB2.open_cursors = 250
```

Description of the illustration init_params_exmpl1.gif

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 = 250` displayed.

Description of the illustration init_params_exmpl2.gif

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.

Description of the illustration init_params_exmpl3.gif

See Also:

- *Oracle Database 2 Day DBA*
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about using a server parameter file in an Oracle Real Application Clusters environment

Administering Storage in Oracle RAC

Most administration tasks for managing storage are the same for Oracle single-instance and Oracle RAC databases. This section provides additional information for using Enterprise Manager to manage some storage structures of an Oracle RAC database.

This section describes the following topics:

- About Automatic Undo Management in Oracle RAC
- Oracle Automatic Storage Management in Oracle RAC
- Administering Redo Logs in Oracle RAC

Name	Status	Host Name	Alerts	Policy Violations	Compliance Score (%)	ASM Instance	ADDM Findings
linuxdb_linuxdb1		Linux1	0 2	0 51 2	98	+ASM1_Linux1 0 2	0
linuxdb_linuxdb2		Linux2	0 2	1 27 1	98	+ASM2_Linux2 0 2	n/a

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.

Select	Name	Allocated Size (MB)	Space Used (MB)	Allocated Space Used(%)	Auto Extend	Allocated Free Space (MB)	Status	Datafiles	Type	Extent Management	Segment Management
◉	SYSAUX	520.0	495.1	95.2	YES	24.9	✓	1	PERMANENT	LOCAL	AUTO
○	SYSTEM	670.0	666.1	99.4	YES	3.9	✓	1	PERMANENT	LOCAL	MANUAL
○	TEMP	30.0	21.0	70.0	YES	9.0	✓	1	TEMPORARY	LOCAL	MANUAL
○	UNDOTBS1	40.0	9.4	23.6	YES	30.6	✓	1	UNDO	LOCAL	MANUAL
○	UNDOTBS2	50.0	8.1	16.2	YES	41.9	✓	1	UNDO	LOCAL	MANUAL
○	USERS	5.0	1.3	26.2	YES	3.7	✓	1	PERMANENT	LOCAL	AUTO

See Also:

- *Oracle Database 2 Day DBA* for more information about managing the undo data for your database

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 does this 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
- About Disk Group Configurations for Oracle ASM in Oracle RAC
- About Standalone Oracle ASM Disk Group Management
- About Oracle ASM Instance and Disk Group Management

Network Listener Configuration

Oracle Net, a component of Oracle Net Services, establishes and maintains a network session from a client application to a database server. After a network session is established, Oracle Net acts as the data courier for both the client application and the database server, exchanging messages between them. Oracle Net can perform these jobs because it is located on each computer in the network.

An important component of Net Services is the Oracle Net Listener (called the listener), which is a separate process that runs on the database server or elsewhere in the network. Client applications can send connection requests to the listener, which manages the traffic of these requests to the database server. When a connection is established, the client and database communicate directly.

The most common ways to configure an Oracle database to service client requests are:

- Dedicated server architecture
  Each client process connects to a dedicated server process. The server process is not shared by any other client for the duration of the client's session. Each new session is assigned a dedicated server process.

- Shared server architecture
  The database uses a pool of shared processes for multiple sessions. A client process communicates with a dispatcher, which is a process that enables many clients to connect to the same database instance without the need for a dedicated server process for each client.

```
[oracle@Linux2 ~]$ srvctl status listener
    Listener LISTENER is enabled
    Listener LISTENER is running on node(s): linux1
[oracle@Linux2 ~]$ srvctl start listener
[oracle@Linux2 ~]$ srvctl status listener
```

```
Listener LISTENER is enabled
Listener LISTENER is running on node(s): linux1,linux2
```

```
[grid@Linux1 admin]$ lsnrctl status
LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 29-APR-2014 11:42:43

Copyright (c) 1991, 2009, Oracle.   All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 11.2.0.1.0 - Production
Start Date                29-APR-2014 09:41:44
Uptime                    0 days 2 hr. 0 min. 59 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /opt/oraproduct/grid/network/admin/listener.ora
Listener Log File         /opt/oraproduct/orabase/diag/tnslsnr/Linux1/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.189.121)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.189.21)(PORT=1521)))
Services Summary...
Service "+ASM" has 1 instance(s).
  Instance "+ASM1", status READY, has 1 handler(s) for this service...
Service "linuxdb" has 1 instance(s).
  Instance "linuxdb1", status READY, has 1 handler(s) for this service...
Service "linuxdbXDB" has 1 instance(s).
  Instance "linuxdb1", status READY, has 1 handler(s) for this service...
The command completed successfully
```

A listener process' behaviour and identity is defined by not just the listening end-points (the 'Host' and 'Port'), but also by the "Listener Parameter File" (listener.ora) which was used to start the listener process. The "Listener Parameter File" defines parameters like, logging and tracing levels, log/trace directories etc. in addition to the listening end-points, and hence, its location is essential for EM to uniquely identify a "Listener Target".

```
[grid@Linux1 admin]$ cat listener.ora
LISTENER=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER))))          # line
added by Agent
LISTENER_SCAN1=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER_SCAN1))))
# line added by Agent
ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER_SCAN1=ON          # line added by Agent
ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER=ON          # line added by Agent
```

[grid@Linux1 admin]$ cat endpoints_listener.ora

LISTENER_LINUX1=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=Linux1-vip)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.189.121)(PORT=1521)(IP=FIRST)))) # line added by Agent

So, we can find 'lsnrctl' use file 'endpoint_listener.ora' to configure and start local listener.

We use 'netca' to configure listener:

$netca

Oracle Net Configuration Assistant: Listener Configuration, Listener
For remote connections to be made to your Oracle database, you must configure a Oracle Net listener. The Oracle Net Configuration Assistant allows you to add, reconfigure, rename or delete a listener.
Select what you want to do:
Add
Reconfigure
Delete
Rename
Cancel
Help
Back
Next

Oracle Net Configuration Assistant: Listener Configuration, Select Listen
Select a listener you want to reconfigure:
LISTENER
Cancel
Help
Back
Next

Oracle Net Configuration Assistant: Listener Configuration, Select Proto
You can configure the listener to listen on a particular subnet and accept connections over one or more protocols. Select the subnet and the protocols you want to configure for this listener. Keep your configuration as simple as possible by configuring only the protocols you need. The subnet information is displayed in the following format: (Network Number) (Subnet Address)/(Subnet Mask)
Select Subnet:
Available Protocols
TCPS
IPC
Selected Protocols
TCP
Cancel
Help
Back
Next

Oracle Net Configuration Assistant: Listener Configuration, TCP/IP Proto
Which TCP/IP port number should the listener use? The port number selected should not be used by any other software on this computer.
Use the standard port number of 1521
Use another port number:
1521
Cancel
Help
Back
Next

Use ‘netmgr’ to assign services to Listener:

$ netmgr

Here, we can add Database services explicitly.

In our tuition, No database services explicitly configured for LISTENER & LISTENTER_SCAN1, all the database services are added (registered) to these listeners while database started, the parameters of database SPFILE , works with tnsnames.ora, defines the listeners IP address and port

[grid@Linux1 ~]$ srvctl add listener -h

Adds a listener configuration to the Oracle Clusterware.

Usage: srvctl add listener [-l <lsnr_name>] [-s] [-p "[TCP:]<port>[, ...][/IPC:<key>][/NMP:<pipe_name>][/TCPS:<s_port>] [/SDP:<port>]"] [-o <oracle_home>] [-k <net_num>]

-l <lsnr_name> Listener name (default name is LISTENER)

-o <oracle_home> ORACLE_HOME path (default value is CRS_HOME)

-k <net_num> VIP network number (default number is 1)

-s Skip the checking of ports

-p "[TCP:]<port>[, ...][/IPC:<key>][/NMP:<pipe_name>][/TCPS:<s_port>] [/SDP:<port>]" Comma separated tcp ports or listener endpoints

-h Print usage

[grid@Linux1 ~]$ srvctl start listener -l LISTENER

Create scond scan listener manually

[grid@Linux1 ~]$ crsctl stat res ora.LISTENER_SCAN1.lsnr -p > ora.LISTENER_SCAN2.lsnr

[grid@Linux1 ~]$ vi ora.LISTENER_SCAN2.lsnr

[grid@Linux1 ~]$ cat ora.LISTENER_SCAN2.lsnr

```
NAME=ora.LISTENER_SCAN2.lsnr
TYPE=ora.scan_listener.type
ACL=owner:grid:rwx,pgrp:oinstall:r-x,other::r--
ACTION_FAILURE_TEMPLATE=
ACTION_SCRIPT=
ACTIVE_PLACEMENT=1
AGENT_FILENAME=%CRS_HOME%/bin/oraagent%CRS_EXE_SUFFIX%
AUTO_START=restore
CARDINALITY=1
CHECK_INTERVAL=60
DEFAULT_TEMPLATE=PROPERTY(RESOURCE_CLASS=scan_listener) PROPERTY(LISTENER_NAME=PARSE(%NAME%, .,
2))
DEGREE=1
DESCRIPTION=Oracle SCAN listener resource
ENABLED=1
ENDPOINTS=TCP:1521
FAILOVER_DELAY=0
FAILURE_INTERVAL=0
FAILURE_THRESHOLD=0
HOSTING_MEMBERS=
LOAD=1
LOGGING_LEVEL=1
NLS_LANG=
NOT_RESTARTING_TEMPLATE=
OFFLINE_CHECK_INTERVAL=0
PLACEMENT=balanced
PORT=1521
PROFILE_CHANGE_TEMPLATE=
RESTART_ATTEMPTS=5
SCRIPT_TIMEOUT=60
SERVER_POOLS=*
START_DEPENDENCIES=hard(ora.scan2.vip) dispersion:active(type:ora.scan_listener.type)
pullup(ora.scan2.vip)
START_TIMEOUT=180
STATE_CHANGE_TEMPLATE=
STOP_DEPENDENCIES=hard(intermediate:ora.scan2.vip)
STOP_TIMEOUT=0
UPTIME_THRESHOLD=1d
USR_ORA_ENV=
USR_ORA_OPI=false
VERSION=11.2.0.1.0
```

[grid@Linux1 ~]$ crsctl add res ora.LISTENER_SCAN2.lsnr -h

```
Usage:
  crsctl add resource <resName> -type <typeName> [[-file <filePath>] | [-attr
"<specification>[,...]"]] [-f] [-i]
    <specification>:   {<attrName>=<value> | <attrName>@<scope>=<value>}
      <scope>:   {@SERVERNAME(<server>)[@DEGREEID(<did>)] |
                  @CARDINALITYID(<cid>)[@DEGREEID(<did>)] }
where
    resName        Add named resource
    typeName       Resource type
    filePath       Attribute file
    attrName       Attribute name
    value         Attribute value
    server         Server name
    cid           Resource cardinality ID
    did           Resource degree ID
    -f            Force option
    -i            Fail if request cannot be processed immediately
```

[grid@Linux1 ~]$ crsctl add res ora.LISTENER_SCAN2.lsnr -type ora.scan_listener.type -file ora.LISTENER_SCAN2.lsnr

[grid@Linux1 ~]$ crsctl stat res

```
NAME=ora.LISTENER.lsnr
TYPE=ora.listener.type
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.LISTENER_SCAN1.lsnr
TYPE=ora.scan_listener.type
TARGET=ONLINE
STATE=ONLINE on linux2

NAME=ora.LISTENER_SCAN2.lsnr
TYPE=ora.scan_listener.type
TARGET=OFFLINE
STATE=OFFLINE

NAME=ora.ORADG.dg
TYPE=ora.diskgroup.type
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.asm
TYPE=ora.asm.type
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.eons
TYPE=ora.eons.type
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.gsd
TYPE=ora.gsd.type
TARGET=OFFLINE, OFFLINE
STATE=OFFLINE, OFFLINE

NAME=ora.linux1.vip
TYPE=ora.cluster_vip_net1.type
TARGET=ONLINE
STATE=ONLINE on linux1

NAME=ora.linux2.vip
TYPE=ora.cluster_vip_net1.type
TARGET=ONLINE
STATE=ONLINE on linux2

NAME=ora.linuxdb.db
TYPE=ora.database.type
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.net1.network
TYPE=ora.network.type
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.oc4j
TYPE=ora.oc4j.type
TARGET=ONLINE
STATE=ONLINE on linux1

NAME=ora.ons
TYPE=ora.ons.type
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.oradg.linuxdblv.acfs
TYPE=ora.acfs.type
TARGET=ONLINE , ONLINE
STATE=OFFLINE, OFFLINE

NAME=ora.oradg.oradg_db.acfs
TYPE=ora.acfs.type
```

```
TARGET=ONLINE , ONLINE
STATE=OFFLINE, OFFLINE

NAME=ora.registry.acfs
TYPE=ora.registry.acfs.type
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.scan1.vip
TYPE=ora.scan_vip.type
TARGET=ONLINE
STATE=ONLINE on linux2

NAME=ora.scan2.vip
TYPE=ora.scan_vip.type
TARGET=ONLINE
STATE=ONLINE on linux1
```

[grid@Linux1 ~]$ crsctl start res ora.LISTENER_SCAN2.lsnr

```
CRS-2672: Attempting to start 'ora.LISTENER_SCAN2.lsnr' on 'linux1'
CRS-2676: Start of 'ora.LISTENER_SCAN2.lsnr' on 'linux1' succeeded
```

[grid@Linux1 ~]$ crsctl stat res

```
NAME=ora.LISTENER.lsnr
TYPE=ora.listener.type
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.LISTENER_SCAN1.lsnr
TYPE=ora.scan_listener.type
TARGET=ONLINE
STATE=ONLINE on linux2

NAME=ora.LISTENER_SCAN2.lsnr
TYPE=ora.scan_listener.type
TARGET=ONLINE
STATE=ONLINE on linux1

NAME=ora.ORADG.dg
TYPE=ora.diskgroup.type
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.asm
TYPE=ora.asm.type
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.eons
TYPE=ora.eons.type
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.gsd
TYPE=ora.gsd.type
TARGET=OFFLINE, OFFLINE
STATE=OFFLINE, OFFLINE

NAME=ora.linux1.vip
TYPE=ora.cluster_vip_net1.type
TARGET=ONLINE
STATE=ONLINE on linux1

NAME=ora.linux2.vip
TYPE=ora.cluster_vip_net1.type
TARGET=ONLINE
STATE=ONLINE on linux2

NAME=ora.linuxdb.db
TYPE=ora.database.type
```

```
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.net1.network
TYPE=ora.network.type
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.oc4j
TYPE=ora.oc4j.type
TARGET=ONLINE
STATE=ONLINE on linux1

NAME=ora.ons
TYPE=ora.ons.type
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.oradg.linuxdblv.acfs
TYPE=ora.acfs.type
TARGET=ONLINE , ONLINE
STATE=OFFLINE, OFFLINE

NAME=ora.oradg.oradg_db.acfs
TYPE=ora.acfs.type
TARGET=ONLINE , ONLINE
STATE=OFFLINE, OFFLINE

NAME=ora.registry.acfs
TYPE=ora.registry.acfs.type
TARGET=ONLINE        , ONLINE
STATE=ONLINE on linux1, ONLINE on linux2

NAME=ora.scan1.vip
TYPE=ora.scan_vip.type
TARGET=ONLINE
STATE=ONLINE on linux2

NAME=ora.scan2.vip
TYPE=ora.scan_vip.type
TARGET=ONLINE
STATE=ONLINE on linux1
```

SCAN Listener

[grid@Linux1 ~]$ lsnrctl service listener_scan1 ; lsnrctl service listener_scan2

```
LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 30-APR-2014 15:07:13

Copyright (c) 1991, 2009, Oracle.  All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER_SCAN2)))
Services Summary...
Service "linuxdb" has 2 instance(s).
  Instance "linuxdb1", status READY, has 1 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:0 refused:0 state:ready
         REMOTE SERVER
         (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=Linux1-vip)(PORT=1521)))
  Instance "linuxdb2", status READY, has 1 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:0 refused:0 state:ready
         REMOTE SERVER
         (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=Linux2-vip)(PORT=1521)))
Service "linuxdbXDB" has 2 instance(s).
  Instance "linuxdb1", status READY, has 1 handler(s) for this service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022 state:ready
         DISPATCHER <machine: Linux1, pid: 6422>
         (ADDRESS=(PROTOCOL=tcp)(HOST=Linux1)(PORT=28167))
```

```
  Instance "linuxdb2", status READY, has 1 handler(s) for this service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022 state:ready
         DISPATCHER <machine: Linux2, pid: 6234>
         (ADDRESS=(PROTOCOL=tcp)(HOST=Linux2)(PORT=58666))
The command completed successfully
```

Local Listener

[grid@Linux1 ~]$ lsnrctl service

```
LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 30-APR-2014 15:07:37

Copyright (c) 1991, 2009, Oracle.  All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER)))
Services Summary...
Service "+ASM" has 1 instance(s).
  Instance "+ASM1", status READY, has 1 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:0 refused:0 state:ready
         LOCAL SERVER
Service "linuxdb" has 1 instance(s).
  Instance "linuxdb1", status READY, has 1 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:0 refused:0 state:ready
         LOCAL SERVER
Service "linuxdbXDB" has 1 instance(s).
  Instance "linuxdb1", status READY, has 1 handler(s) for this service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022 state:ready
         DISPATCHER <machine: Linux1, pid: 6422>
         (ADDRESS=(PROTOCOL=tcp)(HOST=Linux1)(PORT=28167))
The command completed successfully
```

[grid@Linux1 ~]$ lsnrctl service listener

```
LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 30-APR-2014 15:07:51

Copyright (c) 1991, 2009, Oracle.  All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER)))
Services Summary...
Service "+ASM" has 1 instance(s).
  Instance "+ASM1", status READY, has 1 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:0 refused:0 state:ready
         LOCAL SERVER
Service "linuxdb" has 1 instance(s).
  Instance "linuxdb1", status READY, has 1 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:0 refused:0 state:ready
         LOCAL SERVER
Service "linuxdbXDB" has 1 instance(s).
  Instance "linuxdb1", status READY, has 1 handler(s) for this service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022 state:ready
         DISPATCHER <machine: Linux1, pid: 6422>
         (ADDRESS=(PROTOCOL=tcp)(HOST=Linux1)(PORT=28167))
The command completed successfully
[grid@Linux1 ~]$
```

Starting and Stopping Oracle RAC Database Instances and Databases

Typically, you start and shut down the cluster database from the Enterprise Manager Cluster Database Home page. By using this 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. This 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.

To start and stop an entire Oracle RAC database

Assuming you are using a server parameter file (SPFILE)

1. Go to the following URL and log in to Enterprise Manager:
   `http://hostname:portnumber/em`
   For example, http://linux1:1158/em.

2. On 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.

   The Startup/Shutdown: Confirmation page appears.

5. Click **Yes**.

To start and stop individual instances, go to the Startup/Shutdown: Select Operation page and 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)**.

Practice:

About Converting an Oracle Database to an Oracle RAC Database

You can use `rconfig`, or Oracle Enterprise Manager to assist you with the task of converting a single-instance database installation to an Oracle Real Application Clusters (Oracle RAC) database. `rconfig` is a command line utility. The Convert to Cluster Database option in Oracle Enterprise Manager Grid 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.

This section contains the following topics:

- About Preparing for Database Conversion
- Overview of the Database Conversion Process Using Grid Control
- Overview of the Database Conversion Process Using **`rconfig`**
- Converting an Oracle RAC Database into 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:

- The existing database and the target Oracle RAC database must be on the same release of Oracle Database 11*g* 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 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 11*g* database products or options, refer to the product documentation library, which is available in the DOC directory on the 11*g* release 2 (11.2) installation media, or on the OTN Web site at
`http://www.oracle.com/technetwork/indexes/documentation/`

Overview of the Database Conversion Process Using Grid Control

The Server Control (SRVCTL) utility is installed on each node by default. You can use SRVCTL to start and stop the database and instances, manage configuration information, and to move or remove instances and services. You can also use SRVCTL to add services. SRVCTL also manages configuration information.

Some SRVCTL operations store configuration information in the Oracle Cluster Registry (OCR). SRVCTL performs other operations, such as starting and stopping instances, by sending requests to the Oracle Clusterware process (CRSD), which then starts or stops the Oracle Clusterware resources.

To use SRVCTL, enter the srvctl command and its options in case sensitive syntax as described under the heading "SRVCTL Command Reference".

```
[grid@Linux2 ~]$ srvctl stop nodeapps -n linux2 -f
    PRKO-2426 : ONS is already stopped on node(s): linux2
    PRKO-2427 : eONS is already stopped on node(s): linux2
    PRKO-2425 : VIP is already stopped on node(s): linux2

[grid@Linux2 ~]$ srvctl stop nodeapps -n linux1 -f
[grid@Linux2 ~]$ srvctl start nodeapps -n linux1
[grid@Linux2 ~]$ srvctl start nodeapps -n linux2

[grid@Linux1 ~]$ srvctl config scan
SCAN name: Linux-cluster, Network: 1/192.168.189.0/255.255.255.0/eth0
SCAN VIP name: scan1, IP: /linux-cluster.lii02.livun.com/192.168.189.100

[grid@Linux1 ~]$ srvctl config scan_listener
SCAN Listener LISTENER_SCAN1 exists. Port: TCP:1521
```

This section summarizes the process of converting a single-instance database to an Oracle RAC database using Oracle Enterprise Manager Grid Control:

- 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.
- Go to the Server subpage and 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 Installation Guide for Linux and UNIX, or for a different platform, for a complete description of this process

Overview of the Database Conversion Process Using `rconfig`

The following list provides an outline of the process of converting a single-instance database to an Oracle RAC database using the `rconfig` utility:

- 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.
  - The database being converted has been backed up successfully.
- Using `rconfig`, you can convert a single-instance database to either an administrator-managed cluster database or a policy-managed cluster database. Modify the parameters in either the `ConvertToRAC_AdminManaged.xml` or `ConvertToRAC_PolicyManaged.xml` sample file, as appropriate for your environment, then save the file to a new location. Both files are located in the *`Oracle_home`*`/assistants/rconfig/sampleXMLs` directory.
- Run the `rconfig` command, supplying the name of the modified XML file as input.
- Complete the post-conversion tasks.

You can also use the `rconfig` utility to convert single-instance Oracle ASM to clustered Oracle ASM.

See Also:
Oracle Real Application Clusters Installation Guide for Linux and UNIX, or for a different platform, for a complete description of this process

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

After you use the `rconfig` utility to 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. This functionality is available starting with Oracle Database 11*g* Release 2 (11.2.0.2)

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

```
srvctl convert database -d <database_name> -c 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.

Administering Database Instances and Cluster Databases

Web-based Oracle Enterprise Manager Database Control and Grid Control interfaces let you manage Oracle Real Application Clusters (Oracle RAC) databases. The Enterprise Manager console is a central point of control for the Oracle environment. Use the Database Control console to initiate cluster database management tasks. Use the Grid Control console to administer multiple Oracle RAC databases and cluster nodes.

This chapter describes how to administer your Oracle RAC environment. It explains the startup and shutdown tasks for database components and how to administer parameters and parameter files in Oracle RAC. This chapter includes the following sections:

- About Oracle Real Application Clusters Database Management
- About Oracle RAC One Node Database Management
- About Oracle RAC Management Using Enterprise Manager
- Starting and Stopping Oracle RAC Databases and Database Instances
- About Oracle Real Application Clusters Initialization Parameters
- About Administering Storage in Oracle RAC

About Oracle Real Application Clusters Database Management

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 computer that is a member of a cluster, or node, 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.

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.

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.

See Also:

- *Oracle Database 2 Day DBA*
- Chapter 9, " Adding and Deleting Nodes and Instances"

About Oracle RAC One Node Database Management

Caution:
By default, any named user may create a server pool. To restrict the operating system users that have this privilege, Oracle strongly recommends that you add specific users to the CRS Administrators list. See *Oracle Clusterware Administration and Deployment Guide* for more information about adding users to the CRS Administrators list.

Oracle RAC One Node databases are administered slightly differently from Oracle RAC or single-instance databases. For 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 and the database and its services will fail over to one of those servers. For 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, for policy-managed Oracle RAC One Node databases, the destination node for online database relocation must be located in the database's server pool.

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.

Practice:

```
[root@Linux1 ~]# su - grid
[grid@Linux1 ~]$ crsctl add crs administrator -u grid
[grid@Linux1 ~]$ crsctl query crs administrator
    CRS Administrator List: grid
```

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

About Oracle RAC Management Using Enterprise Manager

The Web-based Oracle Enterprise Manager Database Control console and the Oracle Enterprise Manager Grid Control console let you 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 Database Control for cluster database management tasks. Use Enterprise Manager Grid Control to administer your entire Oracle RAC environment, not just the Oracle RAC database.

When you log in to Enterprise Manager using a client browser, 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:
Database Control enables management of all the instances for a single Oracle RAC database. Additionally, Database Control manages all the Oracle ASM instances on the nodes. Only one Oracle ASM instance exists per node, no matter how many Oracle databases use Oracle ASM for storage.

See Also:

- *Oracle Database 2 Day DBA*
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about monitoring Oracle RAC performance

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:

- "About Oracle Clusterware"

- "About Workload Management"

Adding and Deleting Nodes and Instances by command line

Add and remove nodes and instances in Oracle Real Application Clusters (Oracle RAC) environments. You can use these methods when configuring a new Oracle RAC environment, or when resizing an existing Oracle RAC environment.

- Preparing the New Node
- Verifying the New Node Meets the Prerequisites for Installation
- Extending the Oracle Grid Infrastructure Home to the New Node
- Extending the Oracle RAC Home Directory
- Adding the New Node to the Cluster using Enterprise Manager
- Creating an Instance on the New Node
- Deleting an Instance From the Cluster Database
- Removing a Node From the Cluster

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

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, you should use `cluvfy` to verify that all the requirements for installation have been met. To verify the new node meets the hardware requirement, runs 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
```

Extending the Oracle Grid Infrastructure Home to the New Node

Now that 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 section 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.

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 `oracle` user (owner of the Oracle Grid Infrastructure for a cluster software installation) on `racnode1`, go to `Grid_home/oui/bin` 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 this command, 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.

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/cluster/product/11.2.0/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.

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 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.

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.

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/oui/bin` directory on `racnode1` and run the `addNode.sh` script in silent mode as shown in the following example:
   ```
   $ cd /u01/app/oracle/product/11.2.0/dbhome_1/oui/bin
   ```

```
$ ./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.

1. 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.

2. 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.

Adding the New Node to the Cluster using 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".

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.

There are two methods of adding an instance to the new node:

- `Adding a New Instance for a Policy-Managed Database`
- `Adding a New Instance for an Administrator-Managed Database`

Note: The steps described in this section require a license for the Enterprise Manager Provisioning Management pack.

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.

1. From the Cluster Database Home page, click Server.

Under the heading Change Database, click Add Instance.

The Add Instance: Cluster Credentials page appears.

2. 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.

3. Modify the `server pool` settings then click Submit.
4. Click the `Database` tab to return to the Cluster Database Home page.

   Review the number of instances available for your cluster 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.

1. From the Cluster Database Home page, click `Server`.
2. Under the heading Change Database, click `Add Instance`.

The Add Instance: Cluster Credentials page appears.

3. Enter the credentials of the Oracle Clusterware software owner and for the SYSASM user, and 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.

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 he failue

9. Click the `Database` tab to return to the Cluster Database Home page.

   The number of instances available in the cluster database is increased by one.

Deleting an Instance from the Cluster Database

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.

You can delete an instance from the cluster using either the Instance Management option of Database Configuration Assistant (DBCA) or using Enterprise Manager. 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`
- `Deleting an Instance From an Administrator-Managed Database`

Note: The steps described in this section require a license for the Enterprise Manager Provisioning Management pack.

To delete an instance from a `policy-managed database`, 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.

1. Log in to Oracle Enterprise Manager Database Control (Database Control) as a SYSDBA user.
2. From the `Cluster Database Home` page, click `Server`.
3. On the Server subpage, under the heading Change Database, click Delete Instance.

The Edit Server Pool page appears.

4. Enter the credentials for the Oracle Database software owner and the SYSASM user, then click Next.

   The Delete Instance: Database Instance page appears

5. Modify the server pool settings then click Submit.

   Click the Database tab to return to the Cluster Database Home page.

6. Review the number of instances available for your cluster 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. Log in to Oracle Enterprise Manager Database Control (Database Control) as a SYSDBA user.
2. From the Cluster Database Home page, click Server.
3. On the Server subpage, under the heading Change Database, click Delete Instance.

   The Delete Instance: Cluster Credentials page appears.

4. Enter the credentials for the Oracle Database software owner and the SYSASM user, then click Next.

   The Delete Instance: Database Instance page appears

5. Select the instance you want to delete, then click Next.

   After the host information has been validated, the Delete Instance: Review page appears.

6. Review the information, and if correct, click Submit Job to continue. Otherwise, click Back and correct the information.

   A Confirmation page appears.

7. Click View Job to view the status of the node deletion job.

   A 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. Click the Cluster tab to view to the Cluster Home page.

   The number of hosts available in the cluster database is reduced by one.

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 11g 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 previous 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.

Viewing the Cluster Database Topology

Using Database Control you can view a graphical representation of your cluster environment. Using the topology view you can quickly see the components that comprise your cluster database environment, such as database instances, listeners, Oracle ASM instances, hosts, and interfaces.

After you click the topology chart to activate the controls, you can mouse over a component to see the status and configuration details for that component. If you select a component in the topology chart, then you can then right-click that component to view a set of menu actions specific for that component.

To view the topology for your cluster database environment:

1. On the Cluster Database Home page, select **Topology**.

   The Topology subpage appears.

Description of the illustration topology_r2.gif

2. (Optional) Move the mouse cursor over any component in the topology diagram to display information about that component in a popup box.
3. Select any component in the topology diagram to change the information displayed in the Selection Details section.
4. (Optional) Click **Legend** at the bottom of the page, on the left-hand side, to display the Topology Legend page.

   This page describes the icons used in Cluster Topology and Cluster Database Topology.

5. (Optional) Right-click the currently selected component to view the menu actions available for that component.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- "Viewing the Cluster Topology Page"
- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"

Monitoring Oracle Clusterware

You can use Enterprise Manager to monitor Oracle Clusterware. Some of the 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

With Oracle Database 11*g* release 2 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 you click the All Metrics link in the Related Links section of the Cluster home page.

This section contains the following topics:

- Accessing the Oracle Clusterware Information
- Reviewing the Oracle Clusterware Home Page
- Viewing the Cluster Performance Page
- Viewing the Cluster Targets Page
- Viewing the Cluster Administration Page
- About the Cluster Interconnects Page
- Viewing the Cluster Topology Page

See Also:

- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- "About Verifying the Oracle Clusterware Installation"
- "Using CRSCTL to Diagnose Cluster Issues"

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. From the Cluster Database Home page, in the General section, click the link next to **Cluster** to view the Cluster Home page.

   Click the **Database** tab to return to the Cluster Database Home page.

2. Under Diagnostic Summary, click the number next to **Interconnect Alerts** to view the Interconnects subpage for the cluster.

   Click the **Database** tab to return to the Cluster Database Home page.

3. In the High Availability section, click the number next to **Problem Services** to display the Cluster Home page.

   Click the **Database** tab to return to the Cluster Database Home page.

4. Select **Topology**. Click a node in the graphical display to activate the controls. Click the **Interface** component. Right-click the Interface component, then choose **View Details** from the menu to display the Interconnects subpage for the cluster.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- "Viewing the Cluster Topology Page"
- "About the Cluster Interconnects Page"

Reviewing the Oracle Clusterware Home Page

The Cluster Home page enables you to monitor the health and workload of your cluster. It provides a central place for general cluster state information and is updated periodically.

The various sections of the Cluster Home page provide information about the cluster environment and status of the hosts, targets, and clusterware components. For example, the Alerts and Diagnostic Summary sections warn you of errors and performance problems that are impacting the operation of your cluster. You can click the provided links to see more detail about the problem areas.

To monitor the general state of the cluster:

1. From the Cluster Database Home page, in the General section, click the link next to **Cluster**.

   The Cluster Home page appears.

Description of the illustration cluster_home_page_r2.gif

2. (Optional) Click the **Refresh** button to update the information displayed.

   The date and time that data was last collected from the cluster is displayed to the left of the Refresh button.

3. Get a quick view of the cluster in the General section, which includes the following information:
   - Status of the cluster, Up or Down

     Click the **Status** link to drill down to cluster availability details.

   - Number of hosts in the cluster
   - Cluster name
   - The status of Oracle Clusterware overall and by host
   - Oracle Clusterware version

- Oracle Clusterware home directory.

4. In the Configuration section, use the **View** list to select which of the following information is displayed for the available hosts in the cluster:
   - Operating Systems (including Hosts and OS Patches)
   - Hardware (including hardware configuration and hosts)

   Click the links under **Host** or **OS Patches** for detailed information.

5. View the Diagnostic Summary section which contains the number of active Interconnect alerts. Click the number of alerts to view the Interconnects subpage.
6. Investigate the Cluster Databases table to view the cluster databases associated with this cluster, their availability, any alerts or policy violations on those databases, their security compliance score, and the database software version.
7. View the Alerts section, which includes the following items:
   - Category list

     Optionally choose a category from the list to view only alerts in that category

   - Critical

     This is the number of metrics that have crossed critical thresholds plus the number of other critical alerts, such as those caused by incidents (critical errors).

   - Warning

     This is the number of metrics that have crossed warning thresholds

   - Alerts table

     The Alerts table provides information about any alerts that have been issued along with the severity rating of each. Click the alert message in the Message column for more information about the alert.

     When an alert is triggered, the name of the metric for which the alert was triggered is displayed in the Name column. The severity icon for the alert (Warning or Critical) is displayed, along with the time the alert was triggered, the value of the alert, and the time the metric's value was last checked.

8. View the date of the Last Security Evaluation and the Compliance score for the cluster in the Security section.

   The compliance score is a value between 0 and 100 where 100 is a state of complete compliance to the security policies. The compliance score calculation for each target and policy combination to a great extent is influenced by the severity of the violation and importance of the policy, and to a lesser extent by the percentage of violating rows over the total number of rows tested.

9. Review the status of any jobs submitted to the cluster within the last seven days in the Job Activity section.
10. Determine if there are patches to be applied to Oracle Clusterware by reviewing the Critical Patch Advisories for Oracle Homes section.

    To view available patches, you must have first configured your My Oracle Support (formerly Oracle*MetaLink*) Credentials as discussed in "Verifying My Oracle Support Credentials".

11. View basic performance statistics for each host in the cluster in the Hosts table at the bottom of the page.

    Click any link in this table to view further details about that statistic.

12. Use the subtabs at the top of the page to view detailed information for Performance, Targets, Interconnects, or Topology.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- "Viewing the Cluster Topology Page"
- "Obtaining the Patch"
- "Verifying Operating System and Software Requirements"

Viewing the Cluster Performance Page

The Cluster Performance page displays utilization statistics, such as CPU, Memory, and Disk I/O, during the past hour for all hosts of a cluster, which is part of the greater Enterprise Manager environment. With this information, you can determine whether you need to add or redistribute resources.

Description of the illustration cluster_perf.gif

Using the charts on the Cluster Performance page, you can:

- View the CPU, Memory, and Disk I/O charts for the cluster across all hosts.
- View the CPU, Memory, and Disk I/O charts for each host individually by clicking the host name in the legend to the right of the chart.

The Cluster Performance page also contains a Hosts table. The Hosts table displays summary information for the hosts for the cluster, their availability, any alerts on those hosts, CPU and memory utilization percentage, and total I/O per second. You can click a host name in the Hosts table to go to the performance page for that host.

Hosts

Name	Status	Clusterware Status	Alerts	Policy Violations	Compliance Score (%)	ASM Instance	CPU Util %	Mem Util %	Total IO/sec
racnode1.example.com			0 0	5 0 0	75	+ASM1_racnode1.example.com 0 0	30.8	60.35	24.2
racnode2.example.com			0 0	5 0 0	82	+ASM2_racnode2.example.com 0 0	7.84	50.1	18.83

Description of the illustration cluster_hosts_table_r2.gif

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- "Using the Cluster Database Performance Page"

Viewing the Cluster Targets Page

The Cluster Targets page provides a complete list of all targets on the cluster. The table includes the target name, type, host, and location, also the target's availability, warning and critical alerts, and last load time.

Cluster: docrac

Latest Data Collected From T

Home | Performance | Targets | Administration | Interconnects | Topology

Name	Host	Oracle Home	Availability	Alerts	Policy Violations
sales.example.com_sales2	racnode2.example.com	/u01/app/oracle/product/11.2.0/dbhome_2		0 2	1 53 2
sales.example.com_sales1	racnode1.example.com	/u01/app/oracle/product/11.2.0/dbhome_2		0 2	0 53 2
sales.example.com	racnode1.example.com *	/u01/app/oracle/product/11.2.0/dbhome_2		0 4	11 106 4
racnode2.example.com	racnode2.example.com	Not Applicable		0 0	5 0 0
racnode1.example.com	racnode1.example.com	Not Applicable		0 0	5 0 0
LISTENER_racnode2.example.com	racnode2.example.com	/u01/grid		0 0	7 5 0

Description of the illustration cluster_targets2_r2.gif

Click a target name to go to the home page for that target. Click the links in the table to get more information about that particular item, alert, or metric.

The Hosts table displays the hosts for the cluster, their availability, any alerts on those hosts, CPU and memory utilization percentage, and total I/O per second.

See Also:
"Viewing the Cluster Database Topology"

Viewing the Cluster Administration Page

Caution:
By default, any named user may create a server pool. To restrict the operating system users that have this privilege, Oracle strongly recommends that you add specific users to the CRS Administrators list. See *Oracle Clusterware Administration and Deployment Guide* for more information about adding users to the CRS Administrators list.

On the Cluster Administration page you can manage Oracle Clusterware resources, create and manage server pools. You can also use this page to configure Oracle Database Quality of Service Management. This functionality is available starting with Oracle Database 11*g* Release 2 (11.2.0.2)

Description of the illustration cluster_admin_r2.gif

If you click Manage Resources, then you go to the Resources page where you can start, stop, or relocate the resources that are registered with Oracle Clusterware.

Description of the illustration cluster_admin_res_r2.gif

See Also:

- "Viewing the Cluster Database Topology"
- *Oracle Database Quality of Service Management User's Guide*

- *Oracle Clusterware Administration and Deployment Guide*

About the Cluster Interconnects Page

The Cluster Interconnects page is useful for monitoring the interconnect interfaces, determining configuration issues, and identifying transfer rate-related issues, including excess traffic. This page helps determine the load added by individual instances and databases on the interconnect. Sometimes you can immediately identify interconnect delays that are due to applications outside Oracle.

You can use this page to perform the following tasks:

- View all interfaces that are configured across the cluster.
- View statistics for the interfaces, such as absolute transfer rates and errors.
- Determine the type of interfaces, such as private or public.
- Determine whether the instance is using a public or private network.
- Determine which database instance is currently using which interface.
- Determine how much the instance is contributing to the transfer rate on the interface.

Description of the illustration cluster_interconnects_r2.gif

The Private Interconnect Transfer Rate value shows a global view of the private interconnect traffic, which is the estimated traffic on all the private networks in the cluster. The traffic is calculated as the summary of the input rate of all private interfaces known to the cluster. For example, if the traffic rate is high, then the values in the Total I/O Rate column in the Interfaces by Hosts table for private interfaces are also high. If the values in this column are high, then you should determine the cause of the high network usage. You can click a number to access the Network Interface Total I/O Rate page for historic statistics and metric values.

Using the Interfaces by Hosts table, you can drill down to the following pages:

- Host Home
- Hardware Details
- Network Interface Total I/O Rate
- Network Interface Total Error Rate

Using the Interfaces in Use by Cluster Databases table, you can view the Total Transfer Rate. This value shows the network traffic generated by individual instances for the interfaces they are using as interconnects. The values indicate how frequently the instances are communicating with other instances.

See Also:

- "Monitoring Oracle Clusterware"
- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- "About Network Hardware Requirements"

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. Individually distinct icons are used 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.

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.

See Also:
"Viewing the Cluster Database Topology"

Troubleshooting Configuration Problems in Oracle RAC Environments

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. The Cluster Ready Services Control (CRSCTL) command `check` enables you to determine the status of several Oracle Clusterware components simultaneously.

This section contains the following topics:

- Using CRSCTL to Diagnose Cluster Issues
- Using the Cluster Verification Utility to Diagnose Problems
- Viewing Oracle RAC Database Alerts
- Viewing Oracle RAC Database Alert Log Messages

See Also:

- "Tools for Installing, Configuring, and Managing Oracle RAC"
- "About Verifying the Oracle Clusterware Installation"

Viewing Oracle RAC Database Alerts

Alert messages are displayed in Enterprise Manager. The Alerts table is similar to that shown for single-instance databases, but in a cluster database, it includes columns for the target name and target type. For example, if a user connected to the `orcl1` instance exceeded their allotted login time, then you would see an alert message with the following values:

- Target name: `orcl_orcl1`
- Target type: Database instance

- Category: Response
- Name: User logon time
- Message: User logon time is 10250 milliseconds
- Alert triggered: *Date and time when the alert condition occurred*

To view the alert messages for an Oracle RAC database:

1. On the Cluster Database Home page, scroll down to the section titled Alerts.

   The section Related Alerts displays nondatabase alert messages, for example, alert messages for Oracle Net.

2. View the alerts for your database and database instances.

   The following screenshot shows an example of the Alerts display for a clustered database named `docrac`.

Alerts

Category All Critical 0 Warnings 6

Previous 1-5 of 6 Next 1

Severity	Target Name	Target Type	Category	Name	Message	Alert Triggered
⚠	docrac_docrac1	Database Instance	Response	User Logon Time (msec)	User logon time is 10250 msecs.	Sep 21, 2006 3:47:30 AM
⚠	docrac	Cluster Database	Invalid Objects by Schema	Owner's Invalid Object Count	3 object(s) are invalid in the SOE1 schema.	Aug 7, 2006 4:18:01 PM
⚠	docrac	Cluster Database	Invalid Objects by Schema	Owner's Invalid Object Count	16 object(s) are invalid in the CC schema.	Aug 7, 2006 4:18:01 PM
⚠	docrac	Cluster Database	Invalid Objects by Schema	Owner's Invalid Object Count	3 object(s) are invalid in the SOE schema.	Aug 7, 2006 4:18:01 PM

Related Alerts

Description of the illustration racalert.gif

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- "Troubleshooting Configuration Problems in Oracle RAC Environments"
- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day + Performance Tuning Guide*

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, scroll down to the Instances section.

2. Click the name of the instance for which you want to view the alert log.

   The Cluster Database Instance Home page appears.

3. In the Diagnostic Summary section, click the link next to the heading **Alert Log** to display the alert log entries containing ORA- errors.

   The Alert Log Errors page appears.

4. (Optional) Click **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*
- "Monitoring Oracle Clusterware"

Upgrading the Oracle Software

An upgrade modifies most, if not all, of the installed Oracle software files, whereas a patch typically modifies only a few files. Upgrades are performed when you install a new software version, or apply a patch set. You can perform rolling upgrades, or you can shut down the Oracle Database and Oracle Clusterware software to perform the upgrade.

With Oracle Clusterware 11*g* release 2, you perform out-of-place upgrades. 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.

Oracle Database release 11.2.0.2 is a full patch set release. To upgrade to Oracle Database release 11.2.0.2, you install the Oracle Grid Infrastructure for a cluster and Oracle Database software into new Oracle home directories instead of applying the patch set to the existing Oracle home directories. This is different from patch set releases for previous Oracle Database releases, where the patch set was always installed in place.

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.

```
[oracle@rac1 ~]$ cat /u01/app/oracle/product/10.2.0/db_1/network/admin/tnsnames.ora
# tnsnames.ora Network Configuration File: /u01/app/oracle/product/10.2.0/db_1/network/admin/tnsn
ames.ora
# Generated by Oracle configuration tools.

TAF =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = rac1-vip)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = rac2-vip)(PORT = 1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = taf)
      (FAILOVER_MODE =
        (TYPE = SELECT)
        (METHOD = BASIC)
        (RETRIES = 180)
        (DELAY = 5)
      )
    )
  )

DEVDB2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = rac2-vip)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = devdb)
```

After the Service created automatically, check tnsnames.ora file and you’ll see that the new entry is added

Try to connect to the database using this service. As you see, we’ll automatically connect to the first instance. Now let’s check the RAC high availability

For this, while connecting to the first instance (devdb1) using a “service”, open new terminal, connect to the first instance and shut it down

Now go back to the first session and query the v$instance view again. As you see, you' ll be automatically forwarded to the second instance

In this step by step tutorial I' ve shown you the deep step by step guide using 150 screenshots to make the RAC installation easier for you. I hope you' ll successfully install RAC and make your own tests. Good Luck!

ADVERTISEMENT

Like this:

Modify the listener.ora for the ASM Instances

The first step is to modify the `listener.ora` file for the ORACLE_HOME running ASM on all nodes in the RAC cluster by adding a new service:

Node 1 - (listener.ora)

```
LISTENER_LINUX1 =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = linux1-vip)(PORT = 1521)(IP = FIRST))
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.100)(PORT = 1521)(IP =
FIRST))
    )
  )

SID_LIST_LISTENER_LINUX1 =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /u01/app/oracle/product/10.2.0/db_1)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (GLOBAL_DBNAME  = +ASM)
      (SID_NAME       = +ASM1)
      (ORACLE_HOME    = /u01/app/oracle/product/10.2.0/db_1)
    )
  )
```

Node 2 - (listener.ora)

```
LISTENER_LINUX2 =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = linux2-vip)(PORT = 1521)(IP = FIRST))
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.101)(PORT = 1521)(IP =
FIRST))
    )
  )

SID_LIST_LISTENER_LINUX2 =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /u01/app/oracle/product/10.2.0/db_1)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (GLOBAL_DBNAME  = +ASM)
```

```
      (SID_NAME        = +ASM2)
      (ORACLE_HOME     = /u01/app/oracle/product/10.2.0/db_1)
    )
  )
```

Add ASM Entries to the tnsnames.ora File on the Client Machine

The next step is to add ASM entries to the **tnsnames.ora** file on the client machine that will be connecting to the ASM instance(s). The client machine in this example is named **alex.idevelopment.info**. A separate tnsnames entry will be created for each ASM instance in the two-node RAC. The two tnsnames entries for this example are named **ORCL1_ASM1** and **ORCL2_ASM2**:

Client Node - (tnsnames.ora)

```
ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = linux1-vip)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = linux2-vip)(PORT = 1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl.idevelopment.info)
    )
  )

ORCL1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = linux1-vip)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl.idevelopment.info)
      (INSTANCE_NAME = orcl1)
    )
  )

ORCL2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = linux2-vip)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl.idevelopment.info)
      (INSTANCE_NAME = orcl2)
    )
  )
```

```
ORCL1_ASM1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = linux1-vip)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = +ASM)
    )
  )

ORCL2_ASM2 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = linux2-vip)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = +ASM)
    )
  )

ORCL_TAF =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = linux1-vip)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = linux2-vip)(PORT = 1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl_taf.idevelopment.info)
      (FAILOVER_MODE =
        (TYPE = SELECT)
        (METHOD = BASIC)
        (RETRIES = 180)
        (DELAY = 5)
      )
    )
  )
```

Bounce the TNS Listener

The final step is to bounce the Oracle TNS Listener. Remember that the listener process will need to be bounced on both of the nodes in the RAC cluster:

```
# -----------
```

```
# FROM linux1
# -----------
[oracle@linux1 ~]$ lsnrctl stop LISTENER_LINUX1
[oracle@linux1 ~]$ lsnrctl start LISTENER_LINUX1

# -----------
# FROM linux2
# -----------
[oracle@linux2 ~]$ lsnrctl stop LISTENER_LINUX2
[oracle@linux2 ~]$ lsnrctl start LISTENER_LINUX2
```

After restarting the TNS listener, the new service should be available from both nodes in the RAC cluster:

```
# -----------
# FROM linux1
# -----------
[oracle@linux1 ~]$ lsnrctl status LISTENER_LINUX1 | grep ASM
Service "+ASM" has 2 instance(s).
  Instance "+ASM1", status UNKNOWN, has 1 handler(s) for this service...
  Instance "+ASM1", status BLOCKED, has 1 handler(s) for this service...
Service "+ASM_XPT" has 1 instance(s).
  Instance "+ASM1", status BLOCKED, has 1 handler(s) for this service...

# -----------
# FROM linux2
# -----------
[oracle@linux2 ~]$ lsnrctl status LISTENER_LINUX2 | grep ASM
Service "+ASM" has 2 instance(s).
  Instance "+ASM2", status UNKNOWN, has 1 handler(s) for this service...
  Instance "+ASM2", status BLOCKED, has 1 handler(s) for this service...
Service "+ASM_XPT" has 1 instance(s).
  Instance "+ASM2", status BLOCKED, has 1 handler(s) for this service...
```

Test Access to ASM from the Client Machine

After the new ASM service(s) have been added and the TNS listener successfully restarted, test access to the ASM instances from the client machine using SQL*Plus:

```
ORA10G on alex: sqlplus "sys/<ASM_SYS_PASSWORD>@orcl1_asm1 as sysdba"
@asm_diskgroups.sql

Disk Group            Sector   Block   Allocation
Name                    Size    Size    Unit Size State       Type   Total
```

```
Size (MB) Used Size (MB) Pct. Used
-------------------- ------- ------- ------------ ----------- ------
--------------- -------------- ---------
FLASH_RECOVERY_AREA      512   4,096    1,048,576 MOUNTED     EXTERN
596,985            339       .06
ORCL_DATA1               512   4,096    1,048,576 MOUNTED     EXTERN
597,017         40,784      6.83

--------------- --------------
Grand Total:
1,194,002         41,123



ORA10G on alex: sqlplus "sys/<ASM_SYS_PASSWORD>@orcl2_asm2 as sysdba"
@asm_diskgroups.sql

Disk Group            Sector   Block   Allocation
Name                    Size    Size    Unit Size State       Type   Total
Size (MB) Used Size (MB) Pct. Used
-------------------- ------- ------- ------------ ----------- ------
--------------- -------------- ---------
FLASH_RECOVERY_AREA      512   4,096    1,048,576 MOUNTED     EXTERN
596,985            339       .06
ORCL_DATA1               512   4,096    1,048,576 MOUNTED     EXTERN
597,017         40,784      6.83

--------------- --------------
Grand Total:
1,194,002         41,123
```

To automate database startup and shutdown using the `dbstart` and `dbshut` scripts on AIX, HP-UX, Linux, Solaris, or Tru64 UNIX, follow these steps:

1. Log in as the `root` user.
2. Open the `oratab` file in any text editor:
   - Solaris:
   - `# vi /var/opt/oracle/oratab`
   -
   - Other operating systems:
   - `# vi /etc/oratab`
   -

   Database entries in the `oratab` file appear in the following format:

   SID:ORACLE_HOME:{Y|N}

In this example, Y or N specifies whether you want the scripts to start up or shut down the database. For each database that you want to start up and shut down, find the instance identifier (SID) for that database, identified by the *SID* in the first field, and change the last field for each to Y.

> Note:
>
> If you add new database instances to the system, remember to edit the entries for those instances in the `oratab` file if you want them to start automatically.

3. Change directory to one of the following directories, depending on your platform:

Platform	Initialization File Directory
AIX	`/etc`
Linux and Solaris	`/etc/init.d`
HP-UX and Tru64 UNIX	`/sbin/init.d`

4.
5. Create a file called `dbora` with the following contents. Change the value specified for the ORACLE_HOME environment variable to an Oracle home directory for your installation, and change the value of the ORACLE environment variable to the user name of the owner of the database installed in the Oracle home directory (typically `oracle`):

```
6.  #! /bin/sh  -x
7.  #
8.  # Change the value of ORACLE_HOME to specify the correct Oracle
    home
9.  # directory for you installation
10.
11. ORACLE_HOME=/u01/app/oracle/product/10.1.0/db_1
12. #
13. # change the value of ORACLE to the login name of the
14. # oracle owner at your site
15. #
16.
17. ORACLE=oracle
18.
19. PATH=${PATH}:$ORACLE_HOME/bin
```

```
20. HOST=`hostname`
21. PLATFORM=`uname`
22. export ORACLE_HOME PATH
23. #
24. if [ ! "$2" = "ORA_DB" ] ; then
25.    if [ "$PLATFORM" = "HP-UX" ] ; then
26.       remsh $HOST -l $ORACLE -n "$0 $1 ORA_DB"
27.       exit
28.    else
29.       rsh $HOST -l $ORACLE  $0 $1 ORA_DB
30.       exit
31.    fi
32. fi
33. #
34. LOG=$ORACLE_HOME/startup.log
35. touch $LOG
36. chmod a+r $LOG
37. #
38.
39. case $1 in
40. 'start')
41.         echo "$0: starting up" >> $LOG
42.         date >> $LOG
43.         # Start Oracle Net
44.         if [ -f $ORACLE_HOME/bin/tnslsnr ] ; then
45.              echo "starting Oracle Net Listener"
46.              $ORACLE_HOME/bin/lsnrctl start >> $LOG 2>&1 &
47.         fi
48.         echo "Starting Oracle databases"
49.         $ORACLE_HOME/bin/dbstart >> $LOG 2>&1 &
50.         ;;
51. 'stop')
52.         echo "$0: shutting down" >> $LOG
53.         date >> $LOG
54.         # Stop Oracle Net
55.         if [ -f $ORACLE_HOME/bin/tnslsnr ] ; then
56.              echo "stopping Oracle Net Listener"
57.              $ORACLE_HOME/bin/lsnrctl stop >> $LOG 2>&1 &
58.         fi
59.         echo "stopping Oracle databases"
60.         $ORACLE_HOME/bin/dbshut >> $LOG 2>&1 &
61.         ;;
62. *)
63.         echo "usage: $0 {start|stop}"
64.         exit
65.         ;;
```

```
esac
#
exit
```

> Note:
>
> This script works only if a password has not been set for the Oracle Net listener. If a password has been set for the listener, this script cannot stop the listener. Also, if the listener name is not the default name LISTENER, you must specify the listener name in the stop and start commands:
> `$ORACLE_HOME/bin/lsnrctl {start|stop} listener_name`

69. Change the group of the `dbora` file to the OSDBA group (typically `dba`) and set the permissions to 750:

```
# chgrp dba dbora
# chmod 750 dbora
```

73. Create symbolic links to the `dbora` script in the appropriate run-level script directories, as follows:

Platform	Symbolic Links Commands
AIX	`# ln -s /etc/dbora /etc/rc.d/rc2.d/S99dbora` `# ln -s /etc/dbora /etc/rc.d/rc2.d/K01dbora`
HP-UX	`# ln -s /sbin/init.d/dbora /sbin/rc3.d/S990dbora` `# ln -s /sbin/init.d/dbora /sbin/rc3.d/K001dbora`

Linux	`# ln -s /etc/init.d/dbora /etc/rc.d/rc3.d/K01dbora`
	`# ln -s /etc/init.d/dbora /etc/rc.d/rc3.d/S99dbora`
	`# ln -s /etc/init.d/dbora /etc/rc.d/rc5.d/K01dbora`
	`# ln -s /etc/init.d/dbora /etc/rc.d/rc5.d/S99dbora`

Solaris	`# ln -s /etc/init.d/dbora /etc/rc3.d/K01dbora` `# ln -s /etc/init.d/dbora /etc/rc3.d/S99dbora`
Tru64 UNIX	`# ln -s /sbin/init.d/dbora /sbin/rc3.d/S99dbora` `# ln -s /sbin/init.d/dbora /sbin/rc3.d/K01dbora`