

## Build Your Own Oracle RAC 11g Cluster on Oracle Enterprise Linux and iSCSI (Continued)

**The information in this guide is not validated by Oracle, is not supported by Oracle, and should only be used at your own risk; it is for educational purposes only.**

## 13. Create Job Role Separation Operating System Privileges Groups, Users, and Directories

*Perform the following user, group, directory configuration, and setting shell limit tasks for the `grid` and `oracle` users on both Oracle RAC nodes in the cluster.*

This section provides the instructions on how to create the operating system users and groups to install all Oracle software using a *Job Role Separation* configuration. The commands in this section should be performed on both Oracle RAC nodes as `root` to create these groups, users, and directories. Note that the group and user IDs must be identical on both Oracle RAC nodes in the cluster. Check to make sure that the group and user IDs you want to use are available on each cluster member node, and confirm that the primary group for each grid infrastructure for a cluster installation owner has the same name and group ID which for the purpose of this guide is `oinstall` (GID 1000).

A Job Role Separation privileges configuration of Oracle is a configuration with operating system groups and users that divide administrative access privileges to the Oracle grid infrastructure installation from other administrative privileges users and groups associated with other Oracle installations (e.g. the Oracle Database software). Administrative privileges access is granted by membership in separate operating system groups, and installation privileges are granted by using different installation owners for each Oracle installation.

One OS user will be created to own each Oracle software product — "`grid`" for the Oracle grid infrastructure owner and "`oracle`" for the Oracle RAC software. Throughout this article, a user created to own the Oracle grid infrastructure binaries is called the `grid` user. This user will own both the Oracle Clusterware and Oracle Automatic Storage Management binaries. The user created to own the Oracle Database binaries (Oracle RAC) will be called the `oracle` user. Both Oracle software owners must have the Oracle Inventory group (`oinstall`) as their primary group, so that each Oracle software installation owner can write to the central inventory (`oraInventory`), and so that OCR and Oracle Clusterware resource permissions are set correctly. The Oracle RAC software owner must also have the OSDBA group and the optional OSOPER group as secondary groups.

This type of configuration is optional but highly recommend by Oracle for organizations that need to restrict user access to Oracle software by responsibility areas for different administrator users. For example, a small organization could simply allocate operating system user privileges so that you can use one administrative user and one group for operating system authentication for all system privileges on the storage and database tiers. With this type of configuration, you can designate the `oracle` user to be the sole installation owner for all Oracle software (Grid infrastructure and the Oracle database software), and designate `oinstall` to be the single group whose members are granted all system privileges for Oracle Clusterware, Automatic Storage Management, and all Oracle Databases on the servers, and all privileges as installation owners. Other organizations, however, have specialized system roles who will be responsible for installing the Oracle software such as system administrators, network administrators, or storage administrators. These different administrator users can configure a system in preparation for an Oracle grid infrastructure for a cluster installation, and complete all configuration tasks that require operating system `root` privileges. When grid infrastructure installation and configuration is completed successfully, a system administrator should only need to provide configuration information and to grant access to the database administrator to run scripts as `root` during an Oracle RAC installation.

The following O/S groups will be created:

Description	OS Group Name	OS Users Assigned to this Group	Oracle Privilege	Oracle Group Name
Oracle Inventory and Software Owner	<code>oinstall</code>	<code>grid</code> , <code>oracle</code>		
Oracle Automatic Storage Management Group	<code>asmadmin</code>	<code>grid</code>	<code>SYSASM</code>	<code>OSASM</code>

ASM Database Administrator Group	asmdba	grid, oracle	SYSDBA for ASM	OSDBA for ASM
ASM Operator Group	asmoper	grid	SYSOPER for ASM	OSOPER for ASM
Database Administrator	dba	oracle	SYSDBA	OSDBA
Database Operator	oper	oracle	SYSOPER	OSOPER

- Oracle Inventory Group (typically `oinstall`)

Members of the `oinstall` group are considered the "owners" of the Oracle software and are granted privileges to write to the Oracle central inventory (`oraInventory`). When you install Oracle software on a Linux system for the first time, OUI creates the `/etc/oraInst.loc` file. This file identifies the name of the Oracle Inventory group (by default, `oinstall`), and the path of the Oracle Central Inventory directory.

By default, if an `oraInventory` group does not exist, then the installer lists the primary group of the installation owner for the grid infrastructure for a cluster as the `oraInventory` group. Ensure that this group is available as a primary group for all planned Oracle software installation owners. For the purpose of this guide, the `grid` and `oracle` installation owners must be configured with `oinstall` as their primary group.

- The Oracle Automatic Storage Management Group (typically `asmadmin`)

This is a required group. Create this group as a separate group if you want to have separate administration privilege groups for Oracle ASM and Oracle Database administrators. In Oracle documentation, the operating system group whose members are granted privileges is called the `OSASM` group, and in code examples, where there is a group specifically created to grant this privilege, it is referred to as `asmadmin`.

Members of the `OSASM` group can use SQL to connect to an Oracle ASM instance as `SYSASM` using operating system authentication. The `SYSASM` privilege that was introduced in Oracle ASM 11g release 1 (11.1) is now fully separated from the `SYSDBA` privilege in Oracle ASM 11g Release 2 (11.2). `SYSASM` privileges no longer provide access privileges on an RDBMS instance. Providing system privileges for the storage tier using the `SYSASM` privilege instead of the `SYSDBA` privilege provides a clearer division of responsibility between ASM administration and database administration, and helps to prevent different databases using the same storage from accidentally overwriting each others files. The `SYSASM` privileges permit mounting and dismounting disk groups, and other storage administration tasks.

- The ASM Database Administrator group (OSDBA for ASM, typically `asmdba`)

Members of the ASM Database Administrator group (OSDBA for ASM) is a subset of the `SYSASM` privileges and are granted read and write access to files managed by Oracle ASM. The grid infrastructure installation owner (`grid`) and all Oracle Database software owners (`oracle`) must be a member of this group, and all users with `OSDBA` membership on databases that have access to the files managed by Oracle ASM must be members of the `OSDBA` group for ASM.

- Members of the ASM Operator Group (OSOPER for ASM, typically `asmoper`)

This is an optional group. Create this group if you want a separate group of operating system users to have a limited set of Oracle ASM instance administrative privileges (the `SYSOPER` for ASM privilege), including starting up and stopping the Oracle ASM instance. By default, members of the `OSASM` group also have all privileges granted by the `SYSOPER` for ASM privilege.

To use the ASM Operator group to create an ASM administrator group with fewer privileges than the default `asmadmin` group, then you must choose the Advanced installation type to install the Grid infrastructure software. In this case, OUI prompts you to specify the name of this group. In this guide, this group is `asmoper`.

If you want to have an OSOPER for ASM group, then the grid infrastructure for a cluster software owner (`grid`) must be a member of this group.

- Database Administrator (OSDBA, typically `dba`)

Members of the `OSDBA` group can use SQL to connect to an Oracle instance as `SYSDBA` using operating system authentication. Members of this group can perform critical database administration tasks, such as creating the database and instance startup and shutdown. The default name for this group is `dba`. The `SYSDBA` system privilege allows access to a database instance even when the database is not open. Control of this privilege is totally outside of the database itself.

The `SYSDBA` system privilege should not be confused with the database role `DBA`. The `DBA` role does not include the `SYSDBA` or `SYSOPER` system privileges.

- Database Operator (OSOPER, typically `oper`)

Members of the `OSOPER` group can use SQL to connect to an Oracle instance as `SYSOPER` using operating system authentication. Members of this optional group have a limited set of database administrative privileges such as managing and running backups. The default name for this group is `oper`. The `SYSOPER` system privilege allows access to a database instance even when the database is not open. Control of this privilege is totally outside of the database itself. To use this group, choose the Advanced installation type to install the Oracle database software.

## Create Groups and User for Grid Infrastructure

Lets start this section by creating the recommended OS groups and user for Grid Infrastructure on both Oracle RAC nodes:

```
[root@racnode1 ~]# groupadd -g 1000 oinstall
[root@racnode1 ~]# groupadd -g 1200 asmadmin
[root@racnode1 ~]# groupadd -g 1201 asmdba
[root@racnode1 ~]# groupadd -g 1202 asmoper
[root@racnode1 ~]# useradd -m -u 1100 -g oinstall -G asmadmin,asmdba,asmoper -d /home/grid -s /bin/bash -c "Grid Infrastructure Owner" grid

[root@racnode1 ~]# id grid
uid=1100(grid) gid=1000(oinstall) groups=1000(oinstall),1200(asmadmin),1201(asmdba),1202(asmoper)
```

Set the password for the `grid` account:

```
[root@racnode1 ~]# passwd grid
Changing password for user grid.
New UNIX password: xxxxxxxxxxxx
Retype new UNIX password: xxxxxxxxxxxx
passwd: all authentication tokens updated successfully.
```

## Create Login Script for the grid User Account

Log in to both Oracle RAC nodes as the `grid` user account and create the following login script (`.bash_profile`):

**Note:** When setting the Oracle environment variables for each Oracle RAC node, make certain to assign each RAC node a unique Oracle SID. For this example, I used:

- `racnode1` : `ORACLE_SID=+ASM1`
- `racnode2` : `ORACLE_SID=+ASM2`

```

[root@racnode1 ~]# su - grid

# -----
# .bash_profile
# -----
# OS User:      grid
# Application:  Oracle Grid Infrastructure
# Version:     Oracle 11g release 2
# -----

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

alias ls="ls -FA"

# -----
# ORACLE_SID
# -----
# Specifies the Oracle system identifier (SID)
# for the Automatic Storage Management (ASM) instance
# running on this node.
# Each RAC node must have a unique ORACLE_SID.
# (i.e. +ASM1, +ASM2,...)
# -----
ORACLE_SID=+ASM1; export ORACLE_SID

# -----
# JAVA_HOME
# -----
# Specifies the directory of the Java SDK and Runtime
# Environment.
# -----
JAVA_HOME=/usr/local/java; export JAVA_HOME

# -----
# ORACLE_BASE
# -----
# Specifies the base of the Oracle directory structure
# for Optimal Flexible Architecture (OFA) compliant
# installations. The Oracle base directory for the
# grid installation owner is the location where
# diagnostic and administrative logs, and other logs
# associated with Oracle ASM and Oracle Clusterware
# are stored.
# -----
ORACLE_BASE=/u01/app/grid; export ORACLE_BASE

# -----
# ORACLE_HOME
# -----

```

```
# Specifies the directory containing the Oracle
# Grid Infrastructure software. For grid
# infrastructure for a cluster installations, the Grid
# home must not be placed under one of the Oracle base
# directories, or under Oracle home directories of
# Oracle Database installation owners, or in the home
# directory of an installation owner. During
# installation, ownership of the path to the Grid
# home is changed to root. This change causes
# permission errors for other installations.
# -----
ORACLE_HOME=/u01/app/11.2.0/grid; export ORACLE_HOME

# -----
# ORACLE_PATH
# -----
# Specifies the search path for files used by Oracle
# applications such as SQL*Plus. If the full path to
# the file is not specified, or if the file is not
# in the current directory, the Oracle application
# uses ORACLE_PATH to locate the file.
# This variable is used by SQL*Plus, Forms and Menu.
# -----
ORACLE_PATH=/u01/app/oracle/common/oracle/sql; export ORACLE_PATH

# -----
# SQLPATH
# -----
# Specifies the directory or list of directories that
# SQL*Plus searches for a login.sql file.
# -----
SQLPATH=/u01/app/common/oracle/sql; export SQLPATH

# -----
# ORACLE_TERM
# -----
# Defines a terminal definition. If not set, it
# defaults to the value of your TERM environment
# variable. Used by all character mode products.
# -----
ORACLE_TERM=xterm; export ORACLE_TERM

# -----
# NLS_DATE_FORMAT
# -----
# Specifies the default date format to use with the
# TO_CHAR and TO_DATE functions. The default value of
# this parameter is determined by NLS_TERRITORY. The
# value of this parameter can be any valid date
# format mask, and the value must be surrounded by
# double quotation marks. For example:
#
```

```

#           NLS_DATE_FORMAT = "MM/DD/YYYY"
#
# -----
NLS_DATE_FORMAT="DD-MON-YYYY HH24:MI:SS"; export NLS_DATE_FORMAT

# -----
# TNS_ADMIN
# -----
# Specifies the directory containing the Oracle Net
# Services configuration files like listener.ora,
# tnsnames.ora, and sqlnet.ora.
# -----
TNS_ADMIN=$ORACLE_HOME/network/admin; export TNS_ADMIN

# -----
# ORA_NLS11
# -----
# Specifies the directory where the language,
# territory, character set, and linguistic definition
# files are stored.
# -----
ORA_NLS11=$ORACLE_HOME/nls/data; export ORA_NLS11

# -----
# PATH
# -----
# Used by the shell to locate executable programs;
# must include the $ORACLE_HOME/bin directory.
# -----
PATH=.:${JAVA_HOME}/bin:${PATH}:${HOME}/bin:$ORACLE_HOME/bin
PATH=${PATH}:/usr/bin:/bin:/usr/bin/X11:/usr/local/bin
PATH=${PATH}:/u01/app/common/oracle/bin
export PATH

# -----
# LD_LIBRARY_PATH
# -----
# Specifies the list of directories that the shared
# library loader searches to locate shared object
# libraries at runtime.
# -----
LD_LIBRARY_PATH=$ORACLE_HOME/lib
LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${ORACLE_HOME/oracm/lib}
LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/lib:/usr/lib:/usr/local/lib
export LD_LIBRARY_PATH

# -----
# CLASSPATH
# -----
# Specifies the directory or list of directories that
# contain compiled Java classes.
# -----

```

```

CLASSPATH=$ORACLE_HOME/JRE
CLASSPATH=${CLASSPATH}:$ORACLE_HOME/jlib
CLASSPATH=${CLASSPATH}:$ORACLE_HOME/rdbms/jlib
CLASSPATH=${CLASSPATH}:$ORACLE_HOME/network/jlib
export CLASSPATH

# -----
# THREADS_FLAG
# -----
# All the tools in the JDK use green threads as a
# default. To specify that native threads should be
# used, set the THREADS_FLAG environment variable to
# "native". You can revert to the use of green
# threads by setting THREADS_FLAG to the value
# "green".
# -----
THREADS_FLAG=native; export THREADS_FLAG

# -----
# TEMP, TMP, and TMPDIR
# -----
# Specify the default directories for temporary
# files; if set, tools that create temporary files
# create them in one of these directories.
# -----
export TEMP=/tmp
export TMPDIR=/tmp

# -----
# UMASK
# -----
# Set the default file mode creation mask
# (umask) to 022 to ensure that the user performing
# the Oracle software installation creates files
# with 644 permissions.
# -----
umask 022

```

### Create Groups and User for Oracle Database Software

Next, create the the recommended OS groups and user for the Oracle database software on both Oracle RAC nodes:

```

[root@racnode1 ~]# groupadd -g 1300 dba
[root@racnode1 ~]# groupadd -g 1301 oper
[root@racnode1 ~]# useradd -m -u 1101 -g oinstall -G dba,oper,asmdba -d /home/oracle -s /bin/bash -c "Oracle Software Owner" oracle

[root@racnode1 ~]# id oracle
uid=1101(oracle) gid=1000(oinstall) groups=1000(oinstall),1201(asmdba),1300(dba),1301(oper)

```

Set the password for the `oracle` account:

```
[root@racnode1 ~]# passwd oracle
Changing password for user oracle.
New UNIX password: xxxxxxxxxxxx
Retype new UNIX password: xxxxxxxxxxxx
passwd: all authentication tokens updated successfully.
```

### Create Login Script for the oracle User Account

Log in to both Oracle RAC nodes as the `oracle` user account and create the following login script (`.bash_profile`):

**Note:** When setting the Oracle environment variables for each Oracle RAC node, make certain to assign each RAC node a unique Oracle SID. For this example, I used:

- `racnode1:ORACLE_SID=racdb1`
- `racnode2:ORACLE_SID=racdb2`

```
[root@racnode1 ~]# su - oracle

# -----
# .bash_profile
# -----
# OS User:      oracle
# Application:  Oracle Database Software Owner
# Version:      Oracle 11g release 2
# -----

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

alias ls="ls -FA"

# -----
# ORACLE_SID
# -----
# Specifies the Oracle system identifier (SID) for
# the Oracle instance running on this node.
# Each RAC node must have a unique ORACLE_SID.
# (i.e. racdb1, racdb2,...)
# -----
ORACLE_SID=racdb1; export ORACLE_SID

# -----
# ORACLE_UNQNAME
# -----
# 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.
# Set ORACLE_UNQNAME equal to the database unique
# name.
# -----
ORACLE_UNQNAME=racdb; export ORACLE_UNQNAME

# -----
# JAVA_HOME
# -----
# Specifies the directory of the Java SDK and Runtime
# Environment.
# -----
JAVA_HOME=/usr/local/java; export JAVA_HOME

# -----
# ORACLE_BASE
# -----
# Specifies the base of the Oracle directory structure
# for Optimal Flexible Architecture (OFA) compliant
# database software installations.
# -----
ORACLE_BASE=/u01/app/oracle; export ORACLE_BASE

# -----
# ORACLE_HOME
# -----
# Specifies the directory containing the Oracle
# Database software.
# -----
ORACLE_HOME=$ORACLE_BASE/product/11.2.0/dbhome_1; export ORACLE_HOME

# -----
# ORACLE_PATH
# -----
# Specifies the search path for files used by Oracle
# applications such as SQL*Plus. If the full path to
# the file is not specified, or if the file is not
# in the current directory, the Oracle application
# uses ORACLE_PATH to locate the file.
# This variable is used by SQL*Plus, Forms and Menu.
# -----
ORACLE_PATH=/u01/app/common/oracle/sql; export ORACLE_PATH

# -----
# SQLPATH
# -----
# Specifies the directory or list of directories that
# SQL*Plus searches for a login.sql file.
# -----
SQLPATH=/u01/app/common/oracle/sql; export SQLPATH
```

```

# -----
# ORACLE_TERM
# -----
# Defines a terminal definition. If not set, it
# defaults to the value of your TERM environment
# variable. Used by all character mode products.
# -----
ORACLE_TERM=xterm; export ORACLE_TERM

# -----
# NLS_DATE_FORMAT
# -----
# Specifies the default date format to use with the
# TO_CHAR and TO_DATE functions. The default value of
# this parameter is determined by NLS_TERRITORY. The
# value of this parameter can be any valid date
# format mask, and the value must be surrounded by
# double quotation marks. For example:
#
#         NLS_DATE_FORMAT = "MM/DD/YYYY"
#
# -----
NLS_DATE_FORMAT="DD-MON-YYYY HH24:MI:SS"; export NLS_DATE_FORMAT

# -----
# TNS_ADMIN
# -----
# Specifies the directory containing the Oracle Net
# Services configuration files like listener.ora,
# tnsnames.ora, and sqlnet.ora.
# -----
TNS_ADMIN=$ORACLE_HOME/network/admin; export TNS_ADMIN

# -----
# ORA_NLS11
# -----
# Specifies the directory where the language,
# territory, character set, and linguistic definition
# files are stored.
# -----
ORA_NLS11=$ORACLE_HOME/nls/data; export ORA_NLS11

# -----
# PATH
# -----
# Used by the shell to locate executable programs;
# must include the $ORACLE_HOME/bin directory.
# -----
PATH=.:${JAVA_HOME}/bin:${PATH}:${HOME}/bin:$ORACLE_HOME/bin
PATH=${PATH}:/usr/bin:/bin:/usr/bin/X11:/usr/local/bin
PATH=${PATH}:/u01/app/common/oracle/bin
export PATH

```

```
# -----
# LD_LIBRARY_PATH
# -----
# Specifies the list of directories that the shared
# library loader searches to locate shared object
# libraries at runtime.
# -----
LD_LIBRARY_PATH=$ORACLE_HOME/lib
LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:$ORACLE_HOME/oracm/lib
LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/lib:/usr/lib:/usr/local/lib
export LD_LIBRARY_PATH

# -----
# CLASSPATH
# -----
# Specifies the directory or list of directories that
# contain compiled Java classes.
# -----
CLASSPATH=$ORACLE_HOME/JRE
CLASSPATH=${CLASSPATH}:$ORACLE_HOME/jlib
CLASSPATH=${CLASSPATH}:$ORACLE_HOME/rdbms/jlib
CLASSPATH=${CLASSPATH}:$ORACLE_HOME/network/jlib
export CLASSPATH

# -----
# THREADS_FLAG
# -----
# All the tools in the JDK use green threads as a
# default. To specify that native threads should be
# used, set the THREADS_FLAG environment variable to
# "native". You can revert to the use of green
# threads by setting THREADS_FLAG to the value
# "green".
# -----
THREADS_FLAG=native; export THREADS_FLAG

# -----
# TEMP, TMP, and TMPDIR
# -----
# Specify the default directories for temporary
# files; if set, tools that create temporary files
# create them in one of these directories.
# -----
export TEMP=/tmp
export TMPDIR=/tmp

# -----
# UMASK
# -----
# Set the default file mode creation mask
# (umask) to 022 to ensure that the user performing
```

```
# the Oracle software installation creates files
# with 644 permissions.
# -----
umask 022
```

### Verify That the User nobody Exists

Before installing the software, complete the following procedure to verify that the user `nobody` exists on both Oracle RAC nodes:

1. To determine if the user exists, enter the following command:

```
# id nobody
uid=99(nobody) gid=99(nobody) groups=99(nobody)
```

If this command displays information about the `nobody` user, then you do not have to create that user.

2. If the user `nobody` does not exist, then enter the following command to create it:

```
# /usr/sbin/useradd nobody
```

3. Repeat this procedure on all the other Oracle RAC nodes in the cluster.

### Create the Oracle Base Directory Path

The final step is to configure an Oracle base path compliant with an Optimal Flexible Architecture (OFA) structure and correct permissions. This will need to be performed on both Oracle RAC nodes in the cluster as `root`.

This guide assumes that the `/u01` directory is being created in the root file system. Please note that this is being done for the sake of brevity and is not recommended as a general practice. Normally, the `/u01` directory would be provisioned as a separate file system with either hardware or software mirroring configured.

```
[root@racnode1 ~]# mkdir -p /u01/app/grid
[root@racnode1 ~]# mkdir -p /u01/app/11.2.0/grid
[root@racnode1 ~]# chown -R grid:oinstall /u01
[root@racnode1 ~]# mkdir -p /u01/app/oracle
[root@racnode1 ~]# chown oracle:oinstall /u01/app/oracle
[root@racnode1 ~]# chmod -R 775 /u01
```

At the end of this section, you should have the following on both Oracle RAC nodes:

- An Oracle central inventory group, or `oraInventory` group (`oinstall`), whose members that have the central inventory group as their primary group are granted permissions to write to the `oraInventory` directory.
- A separate OSASM group (`asmadmin`), whose members are granted the `SYSASM` privilege to administer Oracle Clusterware and Oracle ASM.
- A separate OSDBA for ASM group (`asmdba`), whose members include `grid` and `oracle`, and who are granted access to Oracle ASM.

- A separate OSOPER for ASM group (`asmoper`), whose members include `grid`, and who are granted limited Oracle ASM administrator privileges, including the permissions to start and stop the Oracle ASM instance.
- An Oracle grid installation for a cluster owner (`grid`), with the `oraInventory` group as its primary group, and with the `OSASM` (`asmadmin`), `OSDBA for ASM` (`asmdba`) and `OSOPER for ASM` (`asmoper`) groups as secondary groups.
- A separate OSDBA group (`dba`), whose members are granted the `SYSDBA` privilege to administer the Oracle Database.
- A separate OSOPER group (`oper`), whose members include `oracle`, and who are granted limited Oracle database administrator privileges.
- An Oracle Database software owner (`oracle`), with the `oraInventory` group as its primary group, and with the `OSDBA` (`dba`), `OSOPER` (`oper`), and the `OSDBA for ASM group` (`asmdba`) as their secondary groups.
- An OFA-compliant mount point `/u01` owned by `grid:oinstall` before installation.
- An Oracle base for the grid `/u01/app/grid` owned by `grid:oinstall` with 775 permissions, and changed during the installation process to 755 permissions. The grid installation owner Oracle base directory is the location where Oracle ASM diagnostic and administrative log files are placed.
- A Grid home `/u01/app/11.2.0/grid` owned by `grid:oinstall` with 775 (`drwxdrwxr-x`) permissions. These permissions are required for installation, and are changed during the installation process to `root:oinstall` with 755 permissions (`drwxr-xr-x`).
- During installation, OUI creates the Oracle Inventory directory in the path `/u01/app/oraInventory`. This path remains owned by `grid:oinstall`, to enable other Oracle software owners to write to the central inventory.
- An Oracle base `/u01/app/oracle` owned by `oracle:oinstall` with 775 permissions.

### Set Resource Limits for the Oracle Software Installation Users

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`):

Shell Limit	Item in limits.conf	Hard Limit
Maximum number of open file descriptors	<code>nofile</code>	65536
Maximum number of processes available to a single user	<code>nproc</code>	16384
Maximum size of the stack segment of the process	<code>stack</code>	10240

To make these changes, run the following as root:

1. On each Oracle RAC node, add the following lines to the `/etc/security/limits.conf` file (the following example shows the software account owners `oracle` and `grid`):

```
[root@racnode1 ~]# cat >> /etc/security/limits.conf <<EOF
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
EOF
```

2. On each Oracle RAC node, add or edit the following line in the `/etc/pam.d/login` file, if it does not already exist:

```
[root@racnode1 ~]# cat >> /etc/pam.d/login <<EOF
session    required    pam_limits.so
EOF
```

3. Depending on your shell environment, make the following changes to the default shell startup file, to change ulimit setting for all Oracle installation owners (note that these examples show the users `oracle` and `grid`):

For the Bourne, Bash, or Korn shell, add the following lines to the `/etc/profile` file by running the following command:

```
[root@racnode1 ~]# cat >> /etc/profile <<EOF
if [ \${USER} = "oracle" ] || [ \${USER} = "grid" ]; then
    if [ \${SHELL} = "/bin/ksh" ]; then
        ulimit -p 16384
        ulimit -n 65536
    else
        ulimit -u 16384 -n 65536
    fi
fi
umask 022
fi
EOF
```

For the C shell (csh or tcsh), add the following lines to the `/etc/csh.login` file by running the following command:

```
[root@racnode1 ~]# cat >> /etc/csh.login <<EOF
if ( \${USER} == "oracle" || \${USER} == "grid" ) then
    limit maxproc 16384
    limit descriptors 65536
endif
EOF
```

---

## 14. Logging In to a Remote System Using X Terminal

This guide requires access to the console of all machines (Oracle RAC nodes and Openfiler) in order to install the operating system and perform several of the configuration tasks. When managing a very small number of servers, it might make sense to connect each server with its own monitor, keyboard, and mouse in order to access its console. However, as the number of servers to manage increases, this solution becomes unfeasible. A more practical solution would be to configure a dedicated computer which

would include a single monitor, keyboard, and mouse that would have direct access to the console of each machine. This solution is made possible using a Keyboard, Video, Mouse Switch —better known as a [KVM Switch](#).

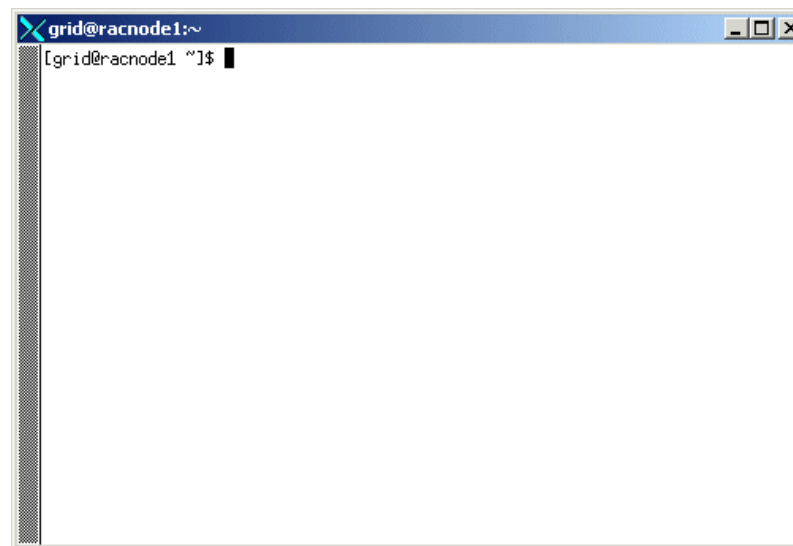
After installing the Linux operating system, there are several applications which are needed to install and configure Oracle RAC which use a Graphical User Interface (GUI) and require the use of an X11 display server. The most notable of these GUI applications (or better known as an X application) is the Oracle Universal Installer (OUI) although others like the Virtual IP Configuration Assistant (VIPCA) also require use of an X11 display server.

Given the fact that I created this article on a system that makes use of a [KVM Switch](#), I am able to toggle to each node and rely on the native X11 display server for Linux in order to display X applications.

If you are not logged directly on to the graphical console of a node but rather you are using a remote client like SSH, PuTTY, or Telnet to connect to the node, any X application will require an X11 display server installed on the client. For example, if you are making a terminal remote connection to `racnode1` from a Windows workstation, you would need to install an X11 display server on that Windows client ([Xming](#) for example). If you intend to install the Oracle grid infrastructure and Oracle RAC software from a Windows workstation or other system with an X11 display server installed, then perform the following actions:

1. Start the X11 display server software on the client workstation.
2. Configure the security settings of the X server software to permit remote hosts to display X applications on the local system.
3. From the client workstation, log in to the server where you want to install the software as the Oracle grid infrastructure for a cluster software owner (`grid`) or the Oracle RAC software (`oracle`).
4. As the software owner (`grid`, `oracle`), set the `DISPLAY` environment:

```
[root@racnode1 ~]# su - grid
[grid@racnode1 ~]$ DISPLAY=<your local workstation>:0.0
[grid@racnode1 ~]$ export DISPLAY
[grid@racnode1 ~]$ # TEST X CONFIGURATION BY RUNNING xterm
[grid@racnode1 ~]$ xterm &
```



**Figure 16:** Test X11 Display Server on Windows; Run xterm from Node 1 (racnode1)

---

## 15. Configure the Linux Servers for Oracle

*Perform the following configuration procedures on both Oracle RAC nodes in the cluster.*

The kernel parameters discussed in this section will need to be defined on both Oracle RAC nodes in the cluster every time the machine is booted. This section provides information about setting those kernel parameters required for Oracle. Instructions for placing them in a startup script (`/etc/sysctl.conf`) is included in [Section 17](#) ("All Startup Commands for Both Oracle RAC Nodes").

### Overview

This section focuses on configuring both Oracle RAC Linux servers - getting each one prepared for the Oracle 11g release 2 grid infrastructure and Oracle RAC 11g release 2 installations on the Oracle Enterprise Linux 5 platform. This includes verifying enough memory and swap space, setting shared memory and semaphores, setting the maximum number of file handles, setting the IP local port range, and finally how to activate all kernel parameters for the system.

There are several different ways to configure (set) these parameters. For the purpose of this article, I will be making all changes permanent (through reboots) by placing all values in the `/etc/sysctl.conf` file.

### Memory and Swap Space Considerations



The minimum required RAM on RHEL/OEL is 1.5 GB for grid infrastructure for a cluster, or 2.5 GB for grid infrastructure for a cluster and Oracle RAC. In this guide, each Oracle RAC node will be hosting Oracle grid infrastructure and Oracle RAC and will therefore require at least 2.5 GB in each server. Each of the Oracle RAC nodes used in this article are equipped with 4 GB of physical RAM.

The minimum required swap space is 1.5 GB. Oracle recommends that you set swap space to 1.5 times the amount of RAM for systems with 2 GB of RAM or less. For systems with 2 GB to 16 GB RAM, use swap space equal to RAM. For systems with more than 16 GB RAM, use 16 GB of RAM for swap space.

- To check the amount of memory you have, type:

```
[root@racnode1 ~]# cat /proc/meminfo | grep MemTotal
MemTotal:      4038564 kB
```

- To check the amount of swap you have allocated, type:

```
[root@racnode1 ~]# cat /proc/meminfo | grep SwapTotal
SwapTotal:      6094840 kB
```

- If you have less than 4GB of memory (between your RAM and SWAP), you can add temporary swap space by creating a temporary swap file. This way you do not have to use a raw device or even more drastic, rebuild your system.

As root, make a file that will act as additional swap space, let's say about 500MB:

```
# dd if=/dev/zero of=tempswap bs=1k count=500000
```

Now we should change the file permissions:

```
# chmod 600 tempswap
```

Finally we format the "partition" as swap and add it to the swap space:

```
# mke2fs tempswap
# mkswap tempswap
# swapon tempswap
```

## Configure Kernel Parameters

The kernel parameters presented in this section are recommended values only as documented by Oracle. For production database systems, Oracle recommends that you tune these values to optimize the performance of the system.

On both Oracle RAC nodes, verify that the kernel parameters described in this section are set to values greater than or equal to the recommended values. Also note that when setting the four semaphore values that all four values need to be entered on one line.

## Configure Kernel Parameters

Oracle Database 11g release 2 on RHEL/OEL 5 requires the kernel parameter settings shown below. The values given are minimums, so if your system uses a larger value,

do not change it.

```
kernel.shmmax = 4294967295
kernel.shmall = 2097152
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.file-max = 6815744
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default=262144
net.core.rmem_max=4194304
net.core.wmem_default=262144
net.core.wmem_max=1048576
fs.aio-max-nr=1048576
```

RHEL/OEL 5 already comes configured with default values defined for the following kernel parameters:

```
kernel.shmall
kernel.shmmax
```

Use the default values if they are the same or larger than the required values.

This article assumes a fresh new install of Oracle Enterprise Linux 5 and as such, many of the required kernel parameters are already set (see above). This being the case, you can simply copy / paste the following to both Oracle RAC nodes while logged in as root:

```
[root@racnode1 ~]# cat >> /etc/sysctl.conf <<EOF

# Controls the maximum number of shared memory segments system wide
kernel.shmmni = 4096

# Sets the following semaphore values:
# SEMMSL_value SEMMNS_value SEMOPM_value SEMMNI_value
kernel.sem = 250 32000 100 128

# Sets the maximum number of file-handles that the Linux kernel will allocate
fs.file-max = 6815744

# Defines the local port range that is used by TCP and UDP
# traffic to choose the local port
net.ipv4.ip_local_port_range = 9000 65500

# Default setting in bytes of the socket "receive" buffer which
# may be set by using the SO_RCVBUF socket option
net.core.rmem_default=262144

# Maximum setting in bytes of the socket "receive" buffer which
# may be set by using the SO_RCVBUF socket option
net.core.rmem_max=4194304

# Default setting in bytes of the socket "send" buffer which
# may be set by using the SO_SNDBUF socket option
net.core.wmem_default=262144
```

```
# Maximum setting in bytes of the socket "send" buffer which
# may be set by using the SO_SNDBUF socket option
net.core.wmem_max=1048576

# Maximum number of allowable concurrent asynchronous I/O requests requests
fs.aio-max-nr=1048576
EOF
```

### Activate All Kernel Parameters for the System

The above command persisted the required kernel parameters through reboots by inserting them in the `/etc/sysctl.conf` startup file. Linux allows modification of these kernel parameters to the current system while it is up and running, so there's no need to reboot the system after making kernel parameter changes. To activate the new kernel parameter values for the currently running system, run the following as `root` on both Oracle RAC nodes in the cluster:

```
[root@racnode1 ~]# sysctl -p
net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.shmmax = 68719476736
kernel.shmall = 4294967296
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.file-max = 6815744
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
fs.aio-max-nr = 1048576
```

Verify the new kernel parameter values by running the following on both Oracle RAC nodes in the cluster:

```
[root@racnode1 ~]# /sbin/sysctl -a | grep shm
vm.hugetlb_shm_group = 0
kernel.shmmni = 4096
kernel.shmall = 4294967296
kernel.shmmax = 68719476736

[root@racnode1 ~]# /sbin/sysctl -a | grep sem
kernel.sem = 250          32000    100      128

[root@racnode1 ~]# /sbin/sysctl -a | grep file-max
fs.file-max = 6815744

[root@racnode1 ~]# /sbin/sysctl -a | grep ip_local_port_range
```

```
net.ipv4.ip_local_port_range = 9000      65500

[root@racnode1 ~]# /sbin/sysctl -a | grep 'core\.[rw]mem'
net.core.rmem_default = 262144
net.core.wmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_max = 1048576
```

---

## 16. Configure RAC Nodes for Remote Access using SSH - (Optional)

*Perform the following optional procedures on both Oracle RAC nodes to manually configure passwordless SSH connectivity between the two cluster member nodes as the "grid" and "oracle" user.*

One of the best parts about this section of the document is that it is completely optional! That's not to say configuring Secure Shell (SSH) connectivity between the Oracle RAC nodes is not necessary. To the contrary, the Oracle Universal Installer (OUI) uses the secure shell tools `ssh` and `scp` commands during installation to run remote commands on and copy files to the other cluster nodes. During the Oracle software installations, SSH must be configured so that these commands do not prompt for a password. The ability to run SSH commands without being prompted for a password is sometimes referred to as *user equivalence*.

The reason this section of the document is optional is that the OUI interface in 11g release 2 includes a new feature that can [automatically configure SSH](#) during the actual install phase of the Oracle software for the user account running the installation. The automatic configuration performed by OUI creates passwordless SSH connectivity between all cluster member nodes. Oracle recommends that you use the automatic procedure whenever possible.

In addition to installing the Oracle software, SSH is used after installation by configuration assistants, Oracle Enterprise Manager, OPatch, and other features that perform configuration operations from local to remote nodes.

**Note:** Configuring SSH with a passphrase is no longer supported for Oracle Clusterware 11g release 2 and later releases. Passwordless SSH is required for Oracle 11g release 2 and higher.

Since this guide uses `grid` as the Oracle grid infrastructure software owner and `oracle` as the owner of the Oracle RAC software, passwordless SSH must be configured for both user accounts.

**Note:** When SSH is not available, the installer attempts to use the `rsh` and `rcp` commands instead of `ssh` and `scp`. These services, however, are disabled by default on most Linux systems. The use of RSH will not be discussed in this article.

### Verify SSH Software is Installed

The supported version of SSH for Linux distributions is OpenSSH. OpenSSH should be included in the Linux distribution minimal installation. To confirm that SSH packages are installed, run the following command on both Oracle RAC nodes:

```
[root@racnode1 ~]# rpm -qa --queryformat "%{NAME}-%{VERSION}-%{RELEASE} (%{ARCH})\n" | grep ssh
openssh-askpass-4.3p2-36.el5 (x86_64)
openssh-clients-4.3p2-36.el5 (x86_64)
openssh-4.3p2-36.el5 (x86_64)
```

```
openssh-server-4.3p2-36.el5 (x86_64)
```

If you do not see a list of SSH packages, then install those packages for your Linux distribution. For example, load CD #1 into each of the Oracle RAC nodes and perform the following to install the OpenSSH packages:

```
[root@racnode1 ~]# mount -r /dev/cdrom /media/cdrom
[root@racnode1 ~]# cd /media/cdrom/Server
[root@racnode1 ~]# rpm -Uvh openssh-*
[root@racnode1 ~]# cd /
[root@racnode1 ~]# eject
```

### Why Configure SSH User Equivalence Using the Manual Method Option?

So, if the OUI already includes a feature that automates the SSH configuration between the Oracle RAC nodes, then why provide a section on how to manually configure passwordless SSH connectivity? In fact, for the purpose of this article, I decided to forgo manually configuring SSH connectivity in favor of Oracle's automatic methods included in the installer.

One reason to include this section on manually configuring SSH is to make mention of the fact that 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. Further documentation on preventing installation errors caused by `stty` commands can be found [later in this section](#).

Another reason you may decide to manually configure SSH for user equivalence is to have the ability to run the Cluster Verification Utility (CVU) prior to installing the Oracle software. The CVU (`runcluvfy.sh`) is a valuable tool located in the Oracle Clusterware root directory that not only verifies all prerequisites have been met before software installation, it also has the ability to generate shell script programs, called fixup scripts, to resolve many incomplete system configuration requirements. The CVU does, however, have a prerequisite of its own and that is SSH user equivalency is configured correctly for the user account running the installation. If you intend to configure SSH connectivity using the OUI, know that the CVU utility will fail before having the opportunity to perform any of its critical checks:

```
[grid@racnode1 ~]$ /media/cdrom/grid/runcluvfy.sh stage -pre crsinst -fixup -n racnode1,racnode2 -verbose
Performing pre-checks for cluster services setup
```

```
Checking node reachability...
```

```
Check: Node reachability from node "racnode1"
```

Destination Node	Reachable?
racnode1	yes
racnode2	yes

```
Result: Node reachability check passed from node "racnode1"
```

```
Checking user equivalence...
```

```
Check: User equivalence for user "grid"
```

Node Name	Comment
racnode2	failed

```
racnode1 failed
Result: PRVF-4007 : User equivalence check failed for user "grid"
```

```
ERROR:
User equivalence unavailable on all the specified nodes
Verification cannot proceed
```

```
Pre-check for cluster services setup was unsuccessful on all the nodes.
```

Please note that it is not required to run the CVU utility before installing the Oracle software. Starting with Oracle 11g release 2, the installer detects when minimum requirements for installation are not completed and performs the same tasks done by the CVU to generate fixup scripts to resolve incomplete system configuration requirements.

### Configure SSH Connectivity Manually on All Cluster Nodes

To reiterate, it is not required to manually configure SSH connectivity before running the OUI. The OUI in 11g release 2 provides an interface during the install for the user account running the installation to automatically create passwordless SSH connectivity between all cluster member nodes. This is the recommend approach by Oracle and the method used in this article. The tasks below to manually configure SSH connectivity between all cluster member nodes is included for documentation purposes only. Keep in mind that this guide uses `grid` as the Oracle grid infrastructure software owner and `oracle` as the owner of the Oracle RAC software. If you decide to manually configure SSH connectivity, it should be performed for both user accounts.

The goal in this section is to setup user equivalence for the `grid` and `oracle` OS user accounts. User equivalence enables the `grid` and `oracle` user accounts to access all other nodes in the cluster (running commands and copying files) without the need for a password. Oracle added support in 10g release 1 for using the SSH tool suite for setting up user equivalence. Before Oracle Database 10g, user equivalence had to be configured using remote shell (RSH).

In the examples that follow, the Oracle software owner listed is the `grid` user.

### Checking Existing SSH Configuration on the System

To determine if SSH is installed and running, enter the following command:

```
[grid@racnode1 ~]$ pgrep sshd
2535
19852
```

If SSH is running, then the response to this command is a list of process ID number(s). Run this command on both Oracle RAC nodes in the cluster to verify the SSH daemons are installed and running.

You need either an RSA or a DSA key for the SSH protocol. RSA is used with the SSH 1.5 protocol, while DSA is the default for the SSH 2.0 protocol. With OpenSSH, you can use either RSA or DSA. The instructions that follow are for SSH1. If you have an SSH2 installation, and you cannot use SSH1, then refer to your SSH distribution documentation to configure SSH1 compatibility or to configure SSH2 with DSA.

**Note:** Automatic passwordless SSH configuration using the OUI creates RSA encryption keys on all nodes of the cluster.

### Configuring Passwordless SSH on Cluster Nodes

To configure passwordless SSH, you must first create RSA or DSA keys on each cluster node, and then copy all the keys generated on all cluster node members into an *authorized keys file* that is identical on each node. Note that the SSH files must be readable only by `root` and by the software installation user (`grid`, `oracle`), as SSH ignores a private key file if it is accessible by others. In the examples that follow, the DSA key is used.

You must configure passwordless SSH separately for each Oracle software installation owner that you intend to use for installation (`grid`, `oracle`).

To configure passwordless SSH, complete the following:

### Create SSH Directory, and Create SSH Keys On Each Node

Complete the following steps on each node:

1. Log in as the software owner (in this example, the `grid` user).

```
[root@racnode1 ~]# su - grid
```

2. To ensure that you are logged in as `grid`, and to verify that the user ID matches the expected user ID you have assigned to the `grid` user, enter the commands `id` and `id grid`. Ensure that Oracle user group and user and the user terminal window process you are using have group and user IDs are identical. For example:

```
[grid@racnode1 ~]$ id
uid=1100(grid) gid=1000(oinstall) groups=1000(oinstall),1200(asmadmin),1201(asmdba),1202(asmoper)
```

```
[grid@racnode1 ~]$ id grid
uid=1100(grid) gid=1000(oinstall) groups=1000(oinstall),1200(asmadmin),1201(asmdba),1202(asmoper)
```

3. If necessary, create the `.ssh` directory in the `grid` user's home directory, and set permissions on it to ensure that only the `grid` user has read and write permissions:

```
[grid@racnode1 ~]$ mkdir ~/.ssh
[grid@racnode1 ~]$ chmod 700 ~/.ssh
```

**Note:** SSH configuration will fail if the permissions are not set to 700.

4. Enter the following command to generate a DSA key pair (public and private key) for the SSH protocol. At the prompts, accept the default key file location and no passphrase (press **[Enter]**):

```
[grid@racnode1 ~]$ /usr/bin/ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/grid/.ssh/id_dsa): [Enter]
Enter passphrase (empty for no passphrase): [Enter]
Enter same passphrase again: [Enter]
Your identification has been saved in /home/grid/.ssh/id_dsa.
Your public key has been saved in /home/grid/.ssh/id_dsa.pub.
The key fingerprint is:
7b:e9:e8:47:29:37:ea:10:10:c6:b6:7d:d2:73:e9:03 grid@racnode1
```

**Note:** SSH with passphrase is not supported for Oracle Clusterware 11g release 2 and later releases. Passwordless SSH is required for Oracle 11g release 2 and higher.

This command writes the DSA public key to the `~/.ssh/id_dsa.pub` file and the private key to the `~/.ssh/id_dsa` file.

Never distribute the private key to anyone not authorized to perform Oracle software installations.

5. Repeat steps 1 through 4 for all remaining nodes that you intend to make a member of the cluster, using the DSA key (`racnode2`).

### Add All Keys to a Common `authorized_keys` File

Now that both Oracle RAC nodes contain a public and private key for DSA, you will need to create an authorized key file (`authorized_keys`) on one of the nodes. An authorized key file is nothing more than a single file that contains a copy of everyone's (every node's) DSA public key. Once the authorized key file contains all of the public keys, it is then distributed to all other nodes in the cluster.

**Note:** The `grid` user's `~/.ssh/authorized_keys` file on every node must contain the contents from all of the `~/.ssh/id_dsa.pub` files that you generated on all cluster nodes.

Complete the following steps on one of the nodes in the cluster to create and then distribute the authorized key file. For the purpose of this article, I am using the primary node in the cluster, `racnode1`:

1. From `racnode1` (the local node) determine if the authorized key file `~/.ssh/authorized_keys` already exists in the `.ssh` directory of the owner's home directory. In most cases this will not exist since this article assumes you are working with a new install. If the file doesn't exist, create it now:

```
[grid@racnode1 ~]$ touch ~/.ssh/authorized_keys
[grid@racnode1 ~]$ ls -l ~/.ssh
total 8
-rw-r--r-- 1 grid oinstall 0 Nov 12 12:34 authorized_keys
-rw----- 1 grid oinstall 668 Nov 12 09:24 id_dsa
-rw-r--r-- 1 grid oinstall 603 Nov 12 09:24 id_dsa.pub
```

In the `.ssh` directory, you should see the `id_dsa.pub` keys that you have created, and the blank file `authorized_keys`.

2. On the local node (`racnode1`), use SCP (Secure Copy) or SFTP (Secure FTP) to copy the content of the `~/.ssh/id_dsa.pub` public key from both Oracle RAC nodes in the cluster to the authorized key file just created (`~/.ssh/authorized_keys`). Again, this will be done from `racnode1`. You will be prompted for the `grid` OS user account password for both Oracle RAC nodes accessed.

The following example is being run from `racnode1` and assumes a two-node cluster, with nodes `racnode1` and `racnode2`:

```
[grid@racnode1 ~]$ ssh racnode1 cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
The authenticity of host 'racnode1 (192.168.1.151)' can't be established.
RSA key fingerprint is 2f:0d:2c:da:9f:d4:3d:2e:ea:e9:98:20:2c:b9:e8:f5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'racnode1,192.168.1.151' (RSA) to the list of known hosts.
grid@racnode1's password: xxxxxx

[grid@racnode1 ~]$ ssh racnode2 cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
The authenticity of host 'racnode2 (192.168.1.152)' can't be established.
RSA key fingerprint is 97:ab:db:26:f6:01:20:cc:e0:63:d0:d1:73:7e:c2:0a.
Are you sure you want to continue connecting (yes/no)? yes
```



```
Warning: Permanently added 'racnode2,192.168.1.152' (RSA) to the list of known hosts.
grid@racnode2's password: xxxxxx
```

The first time you use SSH to connect to a node from a particular system, you will see a message similar to the following:

```
The authenticity of host 'racnode1 (192.168.1.151)' can't be established.
RSA key fingerprint is 2f:0d:2c:da:9f:d4:3d:2e:ea:e9:98:20:2c:b9:e8:f5.
Are you sure you want to continue connecting (yes/no)? yes
```

Enter **yes** at the prompt to continue. The public hostname will then be added to the `known_hosts` file in the `~/.ssh` directory and you will not see this message again when you connect from this system to the same node.

- At this point, we have the DSA public key from every node in the cluster in the authorized key file (`~/.ssh/authorized_keys`) on `racnode1`:

```
[grid@racnode1 ~]$ ls -l ~/.ssh
total 16
-rw-r--r-- 1 grid oinstall 1206 Nov 12 12:45 authorized_keys
-rw----- 1 grid oinstall  668 Nov 12 09:24 id_dsa
-rw-r--r-- 1 grid oinstall  603 Nov 12 09:24 id_dsa.pub
-rw-r--r-- 1 grid oinstall  808 Nov 12 12:45 known_hosts
```

We now need to copy it to the remaining nodes in the cluster. In our two-node cluster example, the only remaining node is `racnode2`. Use the `scp` command to copy the authorized key file to all remaining nodes in the cluster:

```
[grid@racnode1 ~]$ scp ~/.ssh/authorized_keys racnode2:~/.ssh/authorized_keys
grid@racnode2's password: xxxxxx
authorized_keys                                100% 1206      1.2KB/s   00:00
```

- Change the permission of the authorized key file for both Oracle RAC nodes in the cluster by logging into the node and running the following:

```
[grid@racnode1 ~]$ chmod 600 ~/.ssh/authorized_keys
```

## Enable SSH User Equivalency on Cluster Nodes

After you have copied the `authorized_keys` file that contains all public keys to each node in the cluster, complete the steps in this section to ensure passwordless SSH connectivity between all cluster member nodes is configured correctly. In this example, the Oracle grid infrastructure software owner will be used which is named `grid`.

When running the test SSH commands in this section, if you see any other messages or text, apart from the date and host name, then the Oracle installation will fail. If any of the nodes prompt for a password or pass phrase then verify that the `~/.ssh/authorized_keys` file on that node contains the correct public keys and that you have created an Oracle software owner with identical group membership and IDs. Make any changes required to ensure that only the date and host name is displayed when you enter these commands. You should ensure that any part of a login script that generates any output, or asks any questions, is modified so it acts only when the shell is an *interactive* shell.

- On the system where you want to run OUI from (`racnode1`), log in as the `grid` user.

```
[root@racnode1 ~]# su - grid
```

2. If SSH is configured correctly, you will be able to use the `ssh` and `scp` commands without being prompted for a password or pass phrase from the terminal session:

```
[grid@racnode1 ~]$ ssh racnode1 "date;hostname"
Fri Nov 13 09:46:56 EST 2009
racnode1
```

```
[grid@racnode1 ~]$ ssh racnode2 "date;hostname"
Fri Nov 13 09:47:34 EST 2009
racnode2
```

3. Perform the same actions above from the remaining nodes in the Oracle RAC cluster (racnode2) to ensure they too can access all other nodes without being prompted for a password or pass phrase and get added to the `known_hosts` file:

```
[grid@racnode2 ~]$ ssh racnode1 "date;hostname"
The authenticity of host 'racnode1 (192.168.1.151)' can't be established.
RSA key fingerprint is 2f:0d:2c:da:9f:d4:3d:2e:ea:e9:98:20:2c:b9:e8:f5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'racnode1,192.168.1.151' (RSA) to the list of known hosts.
Fri Nov 13 10:19:57 EST 2009
racnode1
```

```
[grid@racnode2 ~]$ ssh racnode1 "date;hostname"
Fri Nov 13 10:20:58 EST 2009
racnode1
```

```
-----

[grid@racnode2 ~]$ ssh racnode2 "date;hostname"
The authenticity of host 'racnode2 (192.168.1.152)' can't be established.
RSA key fingerprint is 97:ab:db:26:f6:01:20:cc:e0:63:d0:d1:73:7e:c2:0a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'racnode2,192.168.1.152' (RSA) to the list of known hosts.
Fri Nov 13 10:22:00 EST 2009
racnode2
```

```
[grid@racnode2 ~]$ ssh racnode2 "date;hostname"
Fri Nov 13 10:22:01 EST 2009
racnode2
```

4. The Oracle Universal Installer is a GUI interface and requires the use of an X Server. From the terminal session enabled for user equivalence (the node you will be performing the Oracle installations from), set the environment variable `DISPLAY` to a valid X Windows display:

Bourne, Korn, and Bash shells:

```
[grid@racnode1 ~]$ DISPLAY=<Any X-Windows Host>:0
[grid@racnode1 ~]$ export DISPLAY
```

C shell:

```
[grid@racnode1 ~]$ setenv DISPLAY <Any X-Windows Host>:0
```

After setting the `DISPLAY` variable to a valid X Windows display, you should perform another test of the current terminal session to ensure that *X11 forwarding* is not enabled:

```
[grid@racnode1 ~]$ ssh racnode1 hostname
racnode1
```

```
[grid@racnode1 ~]$ ssh racnode2 hostname
racnode2
```

**Note:** If you are using a remote client to connect to the node performing the installation, and you see a message similar to: "Warning: No xauth data; using fake authentication data for X11 forwarding." then this means that your authorized keys file is configured correctly; however, your SSH configuration has *X11 forwarding* enabled. For example:

```
[grid@racnode1 ~]$ export DISPLAY=melody:0
[grid@racnode1 ~]$ ssh racnode2 hostname
Warning: No xauth data; using fake authentication data for X11 forwarding.
racnode2
```

Note that having X11 Forwarding enabled will cause the Oracle installation to fail. To correct this problem, create a user-level SSH client configuration file for the oracle OS user account that disables X11 Forwarding:

1. Using a text editor, edit or create the file `~/.ssh/config`
2. Make sure that the `ForwardX11` attribute is set to `no`. For example, insert the following into the `~/.ssh/config` file:

```
Host *
    ForwardX11 no
```

### Preventing Installation Errors Caused by `stty` Commands

During an Oracle grid infrastructure or Oracle RAC software installation, OUI uses SSH to run commands and copy files to the other nodes. During the installation, hidden files on the system (for example, `.bashrc` or `.cshrc`) will cause makefile and other installation errors if they contain `stty` commands.

To avoid this problem, you must modify these files in each Oracle installation owner user home directory to suppress all output on `STDERR`, as in the following examples:

- Bourne, Bash, or Korn shell:

```
if [ -t 0 ]; then
    stty intr ^C
fi
```

- C shell:

```
test -t 0
if ($status == 0) then
    stty intr ^C
endif
```

**Note:** If there are hidden files that contain `stty` commands that are loaded by the remote shell, then OUI indicates an error and stops the installation.

---

## 17. All Startup Commands for Both Oracle RAC Nodes

*Verify that the following startup commands are included on both of the Oracle RAC nodes in the cluster.*

Up to this point, we have talked in great detail about the parameters and resources that need to be configured on both nodes in the Oracle RAC 11g configuration. This section will review those parameters, commands, and entries from previous sections that need to occur on both Oracle RAC nodes when they are booted.

For each of the startup files below, entries in **red** should be included in each startup file.

### **/etc/sysctl.conf**

*We wanted to adjust the default and maximum send buffer size as well as the default and maximum receive buffer size for the interconnect. This file also contains those parameters responsible for configuring shared memory, semaphores, file handles, and local IP range used by the Oracle instance.*

```
.....
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename
# Useful for debugging multi-threaded applications
kernel.core_uses_pid = 1

# Controls the use of TCP syncookies
net.ipv4.tcp_syncookies = 1

# Controls the maximum size of a message, in bytes
kernel.msgmnb = 65536

# Controls the default maximum size of a message queue
```

```

kernel.msgmax = 65536

# Controls the maximum shared segment size, in bytes
kernel.shmmax = 68719476736

# Controls the maximum number of shared memory segments, in pages
kernel.shmall = 4294967296

# Controls the maximum number of shared memory segments system wide
kernel.shmmni = 4096

# Sets the following semaphore values:
# SEMMSL_value SEMMNS_value SEMOPM_value SEMMNI_value
kernel.sem = 250 32000 100 128

# Sets the maximum number of file-handles that the Linux kernel will allocate
fs.file-max = 6815744

# Defines the local port range that is used by TCP and UDP
# traffic to choose the local port
net.ipv4.ip_local_port_range = 9000 65500

# Default setting in bytes of the socket "receive" buffer which
# may be set by using the SO_RCVBUF socket option
net.core.rmem_default=262144

# Maximum setting in bytes of the socket "receive" buffer which
# may be set by using the SO_RCVBUF socket option
net.core.rmem_max=4194304

# Default setting in bytes of the socket "send" buffer which
# may be set by using the SO_SNDBUF socket option
net.core.wmem_default=262144

# Maximum setting in bytes of the socket "send" buffer which
# may be set by using the SO_SNDBUF socket option
net.core.wmem_max=1048576

# Maximum number of allowable concurrent asynchronous I/O requests requests
fs.aio-max-nr=1048576
.....

```

Verify that each of the required kernel parameters are configured in the `/etc/sysctl.conf` file. Then, ensure that each of these parameters are truly in effect by running the following command on both Oracle RAC nodes in the cluster:

```

[root@racnode1 ~]# sysctl -p
net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1

```

```

kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.shmmax = 68719476736
kernel.shmall = 4294967296
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.file-max = 6815744
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
fs.aio-max-nr = 1048576

```

#### **/etc/hosts**

*All machine/IP entries for nodes in our RAC cluster.*

```

.....
# Do not remove the following line, or various programs
# that require network functionality will fail.

127.0.0.1          localhost.localdomain  localhost

# Public Network - (eth0)
192.168.1.151      racnode1
192.168.1.152      racnode2

# Private Interconnect - (eth1)
192.168.2.151      racnode1-priv
192.168.2.152      racnode2-priv

# Public Virtual IP (VIP) addresses - (eth0:1)
192.168.1.251      racnode1-vip
192.168.1.252      racnode2-vip

# Single Client Access Name (SCAN)
192.168.1.187      racnode-cluster-scan

# Private Storage Network for Openfiler - (eth1)
192.168.1.195      openfiler1
192.168.2.195      openfiler1-priv

# Miscellaneous Nodes
192.168.1.1         router
192.168.1.105       packmule
192.168.1.106       melody
192.168.1.121       domo
192.168.1.122       switch1
192.168.1.125       oemprod
192.168.1.245       accesspoint

```

.....

**/etc/udev/rules.d/55-openiscsi.rules**

*Rules file to be used by udev to mount iSCSI volumes. This file contains all name=value pairs used to receive events and the call-out SHELL script to handle the event.*

```
.....
# /etc/udev/rules.d/55-openiscsi.rules
KERNEL=="sd*", BUS=="scsi", PROGRAM="/etc/udev/scripts/iscsidev.sh %b",SYMLINK+="iscsi/%c/part%n"
.....
```

**/etc/udev/scripts/iscsidev.sh**

*Call-out SHELL script that handles the events passed to it from the udev rules file (above) and used to mount iSCSI volumes.*

```
.....
#!/bin/sh

# FILE: /etc/udev/scripts/iscsidev.sh

BUS=${1}
HOST=${BUS%:*}

[ -e /sys/class/iscsi_host ] || exit 1

file="/sys/class/iscsi_host/host${HOST}/device/session*/iscsi_session*/targetname"

target_name=$(cat ${file})

# This is not an open-scsi drive
if [ -z "${target_name}" ]; then
    exit 1
fi

# Check if QNAP drive
check_qnap_target_name=${target_name%:*}
if [ $check_qnap_target_name = "iqn.2004-04.com.qnap" ]; then
    target_name=`echo "${target_name%.*}"`
fi

echo "${target_name##*.*}"
.....
```

---

## 18. Install and Configure ASMLib 2.0

*The installation and configuration procedures in this section should be performed on both of the Oracle RAC nodes in the cluster. Creating the ASM disks, however, will only need to be performed on a single node within the cluster (racnode1).*

In this section, we will install and configure ASMLib 2.0 which is a support library for the Automatic Storage Management (ASM) feature of the Oracle Database. In this article, ASM will be used as the shared file system and volume manager for Oracle Clusterware files (OCR and voting disk), Oracle Database files (data, online redo logs, control files, archived redo logs), and the Fast Recovery Area.

Automatic Storage Management simplifies database administration by eliminating the need for the DBA to directly manage potentially thousands of Oracle database files requiring only the management of groups of disks allocated to the Oracle Database. ASM is built into the Oracle kernel and can be used for both single and clustered instances of Oracle. All of the files and directories to be used for Oracle will be contained in a *disk group* — (or for the purpose of this article, three disk groups). ASM automatically performs load balancing in parallel across all available disk drives to prevent hot spots and maximize performance, even with rapidly changing data usage patterns. ASMLib allows an Oracle Database using ASM more efficient and capable access to the disk groups it is using.

Keep in mind that ASMLib is only a support library for the ASM software. The ASM software will be installed as part of Oracle grid infrastructure [later in this guide](#). Starting with Oracle grid infrastructure 11g release 2 (11.2), the Automatic Storage Management and Oracle Clusterware software is packaged together in a single binary distribution and installed into a single home directory, which is referred to as the Grid Infrastructure home. The Oracle grid infrastructure software will be owned by the user `grid`.

So, is ASMLib required for ASM? Not at all. In fact, there are two different methods to configure ASM on Linux:

- **ASM with ASMLib I/O:** This method creates all Oracle database files on raw *block devices* managed by ASM using ASMLib calls. RAW devices are not required with this method as ASMLib works with block devices.
- **ASM with Standard Linux I/O:** This method does not make use of ASMLib. Oracle database files are created on raw *character devices* managed by ASM using standard Linux I/O system calls. You will be required to create RAW devices for all disk partitions used by ASM.

In this article, I will be using the "ASM with ASMLib I/O" method. Oracle states in Metalink Note 275315.1 that "*ASMLib was provided to enable ASM I/O to Linux disks without the limitations of the standard UNIX I/O API*". I plan on performing several tests in the future to identify the performance gains in using ASMLib. Those performance metrics and testing details are out of scope of this article and therefore will not be discussed.

If you would like to learn more about Oracle ASMLib 2.0, visit <http://www.oracle.com/technology/tech/linux/asmlib/>

## Install ASMLib 2.0 Packages

In previous editions of this article, here would be the time where you would need to download the ASMLib 2.0 software from [Oracle ASMLib Downloads for Red Hat Enterprise Linux Server 5](#). This is no longer necessary since the ASMLib software is included with Oracle Enterprise Linux (*with the exception of the Userspace Library which is a separate download*). The ASMLib 2.0 software stack includes the following packages:

### 32-bit (x86) Installations

- ASMLib Kernel Driver
  - `oracleasm-x.x.x-x.el5-x.x.x-x.el5.i686.rpm` - (for default kernel)
  - `oracleasm-x.x.x-x.el5xen-x.x.x-x.el5.i686.rpm` - (for xen kernel)



- Userspace Library
  - oracleasm-lib-x.x.x-x.el5.i386.rpm
- Driver Support Files
  - oracleasm-support-x.x.x-x.el5.i386.rpm

### **64-bit (x86\_64) Installations**

- ASMLib Kernel Driver
  - oracleasm-x.x.x-x.el5-x.x.x-x.el5.x86\_64.rpm - (for default kernel)
  - oracleasm-x.x.x-x.el5xen-x.x.x-x.el5.x86\_64.rpm - (for xen kernel)
- Userspace Library
  - oracleasm-lib-x.x.x-x.el5.x86\_64.rpm
- Driver Support Files
  - oracleasm-support-x.x.x-x.el5.x86\_64.rpm

With Oracle Enterprise Linux 5, the ASMLib 2.0 software packages do not get installed by default. The ASMLib 2.0 kernel drivers can be found on CD #5 while the Driver Support File can be found on CD #3. The Userspace Library will need to be downloaded as it is not included with Enterprise Linux. To determine if the Oracle ASMLib packages are installed (which in most cases, they will not be), perform the following on both Oracle RAC nodes:

```
[root@racnode1 ~]# rpm -qa --queryformat "%{NAME}-%{VERSION}-%{RELEASE} (%{ARCH})\n" | grep oracleasm | sort
```

If the Oracle ASMLib 2.0 packages are not installed, load the Enterprise Linux CD #3 and then CD #5 into each of the Oracle RAC nodes and perform the following:

#### **From Enterprise Linux 5.4 (x86\_64) - [CD #3]**

```
mount -r /dev/cdrom /media/cdrom
cd /media/cdrom/Server
rpm -Uvh oracleasm-support-2.1.3-1.el5.x86_64.rpm
cd /
eject
```

#### **From Enterprise Linux 5.4 (x86\_64) - [CD #5]**

```
mount -r /dev/cdrom /media/cdrom
cd /media/cdrom/Server
rpm -Uvh oracleasm-2.6.18-164.el5-2.0.5-1.el5.x86_64.rpm
cd /
eject
```

After installing the ASMLib packages, verify from both Oracle RAC nodes that the software is installed:

```
[root@racnode1 ~]# rpm -qa --queryformat "%{NAME}-%{VERSION}-%{RELEASE} (%{ARCH})\n" | grep oracleasm | sort
oracleasm-2.6.18-164.el5-2.0.5-1.el5 (x86_64)
oracleasm-support-2.1.3-1.el5 (x86_64)
```

### **Download Oracle ASMLib Userspace Library**

As mentioned in the previous section, the ASMLib 2.0 software is included with Enterprise Linux with the exception of the Userspace Library (a.k.a. the ASMLib support library). The Userspace Library is required and can be downloaded for free at:

### **32-bit (x86) Installations**

- [oracleasm-lib-2.0.4-1.el5.i386.rpm](#)

### **64-bit (x86\_64) Installations**

- [oracleasm-lib-2.0.4-1.el5.x86\\_64.rpm](#)

After downloading the Userspace Library to both Oracle RAC nodes in the cluster, install it using the following:

```
[root@racnode1 ~]# rpm -Uvh oracleasm-lib-2.0.4-1.el5.x86_64.rpm
Preparing... ##### [100%]
1:oracleasm-lib ##### [100%]
```

For information on obtaining the ASMLib support library through the Unbreakable Linux Network (which is not a requirement for this article), please visit [Getting Oracle ASMLib via the Unbreakable Linux Network](#).

### **Configure ASMLib**

Now that you have installed the ASMLib Packages for Linux, you need to configure and load the ASM kernel module. This task needs to be run on both Oracle RAC nodes as the root user account.

**Note:** The `oracleasm` command by default is in the path `/usr/sbin`. The `/etc/init.d` path, which was used in previous releases, is not deprecated, but the `oracleasm` binary in that path is now used typically for internal commands. If you enter the command `oracleasm configure` without the `-i` flag, then you are shown the current configuration. For example,

```
[root@racnode1 ~]# /usr/sbin/oracleasm configure
ORACLEASM_ENABLED=false
ORACLEASM_UID=
ORACLEASM_GID=
ORACLEASM_SCANBOOT=true
ORACLEASM_SCANORDER=""
ORACLEASM_SCANEXCLUDE=""
```

1. Enter the following command to run the `oracleasm` initialization script with the `configure` option:

```
[root@racnode1 ~]# /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 []: asmadmin
Start Oracle ASM library driver on boot (y/n) [n]: y
Scan for Oracle ASM disks on boot (y/n) [y]: y
Writing Oracle ASM library driver configuration: done

```

The script completes the following tasks:

- o Creates the `/etc/sysconfig/oracleasm` configuration file
- o Creates the `/dev/oracleasm` mount point
- o Mounts the ASMLib driver file system

**Note:** The ASMLib driver file system is not a regular file system. It is used only by the Automatic Storage Management library to communicate with the Automatic Storage Management driver.

2. Enter the following command to load the `oracleasm` kernel module:

```

[root@racnode1 ~]# /usr/sbin/oracleasm init
Creating /dev/oracleasm mount point: /dev/oracleasm
Loading module "oracleasm": oracleasm
Mounting ASMLib driver filesystem: /dev/oracleasm

```

3. Repeat this procedure on all nodes in the cluster (`racnode2`) where you want to install Oracle RAC.

### Create ASM Disks for Oracle

Creating the ASM disks only needs to be performed from one node in the RAC cluster as the `root` user account. I will be running these commands on `racnode1`. On the other Oracle RAC node(s), you will need to perform a *scandisk* to recognize the new volumes. When that is complete, you should then run the `oracleasm listdisks` command on both Oracle RAC nodes to verify that all ASM disks were created and available.

In the section "[Create Partitions on iSCSI Volumes](#)", we configured (partitioned) three iSCSI volumes to be used by ASM. ASM will be used for storing Oracle Clusterware files, Oracle database files like online redo logs, database files, control files, archived redo log files, and the Fast Recovery Area. Use the local device names that were created by `udev` when configuring the three ASM volumes.

To create the ASM disks using the iSCSI target names to local device name mappings, type the following:

```

[root@racnode1 ~]# /usr/sbin/oracleasm createdisk CRSVOL1 /dev/iscsi/crs1/part1
Writing disk header: done
Instantiating disk: done

[root@racnode1 ~]# /usr/sbin/oracleasm createdisk DATAVOL1 /dev/iscsi/data1/part1
Writing disk header: done
Instantiating disk: done

[root@racnode1 ~]# /usr/sbin/oracleasm createdisk FRAVOL1 /dev/iscsi/fra1/part1
Writing disk header: done

```

```
Instantiating disk: done
```

To make the disk available on the other nodes in the cluster (racnode2), enter the following command as `root` on each node:

```
[root@racnode2 ~]# /usr/sbin/oracleasm scandisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
Instantiating disk "FRAVOL1"
Instantiating disk "DATAVOL1"
Instantiating disk "CRSVOL1"
```

We can now test that the ASM disks were successfully created by using the following command on both nodes in the RAC cluster as the `root` user account. This command identifies shared disks attached to the node that are marked as Automatic Storage Management disks:

```
[root@racnode1 ~]# /usr/sbin/oracleasm listdisks
CRSVOL1
DATAVOL1
FRAVOL1
```

```
[root@racnode2 ~]# /usr/sbin/oracleasm listdisks
CRSVOL1
DATAVOL1
FRAVOL1
```

---

## 19. Download Oracle RAC 11g Release 2 Software

*The following download procedures only need to be performed on one node in the cluster.*

The next step is to download and extract the required Oracle software packages from the Oracle Technology Network (OTN):

**Note:** If you do not currently have an account with Oracle OTN, you will need to create one. This is a FREE account!

Oracle offers a development and testing license free of charge. No support, however, is provided and the license does not permit production use. A full description of the license agreement is available on OTN.

### **32-bit (x86) Installations**

[http://www.oracle.com/technology/software/products/database/oracle11g/112010\\_linuxsoft.html](http://www.oracle.com/technology/software/products/database/oracle11g/112010_linuxsoft.html)

### **64-bit (x86\_64) Installations**

[http://www.oracle.com/technology/software/products/database/oracle11g/112010\\_linx8664soft.html](http://www.oracle.com/technology/software/products/database/oracle11g/112010_linx8664soft.html)

You will be downloading and extracting the required software from Oracle to only one of the Linux nodes in the cluster — namely, `racnode1`. You will perform all Oracle software installs from this machine. The Oracle installer will copy the required software packages to all other nodes in the RAC configuration using remote access (`scp`).

Log in to the node that you will be performing all of the Oracle installations from (`racnode1`) as the appropriate software owner. For example, login and download the Oracle grid infrastructure software to the directory `/home/grid/software/oracle` as the `grid` user. Next, log in and download the Oracle Database and Oracle Examples (optional) software to the `/home/oracle/software/oracle` directory as the `oracle` user.

### Download and Extract the Oracle Software

Download the following software packages:

- Oracle Database 11g Release 2 Grid Infrastructure (11.2.0.1.0) for Linux
- Oracle Database 11g Release 2 (11.2.0.1.0) for Linux
- Oracle Database 11g Release 2 Examples (optional)

All downloads are available from the same page.

Extract the Oracle grid infrastructure software as the `grid` user:

```
[grid@racnode1 ~]$ mkdir -p /home/grid/software/oracle
[grid@racnode1 ~]$ mv linux.x64_11gR2_grid.zip /home/grid/software/oracle
[grid@racnode1 ~]$ cd /home/grid/software/oracle
[grid@racnode1 oracle]$ unzip linux.x64_11gR2_grid.zip
```

Extract the Oracle Database and Oracle Examples software as the `oracle` user:

```
[oracle@racnode1 ~]$ mkdir -p /home/oracle/software/oracle
[oracle@racnode1 ~]$ mv linux.x64_11gR2_database_1of2.zip /home/oracle/software/oracle
[oracle@racnode1 ~]$ mv linux.x64_11gR2_database_2of2.zip /home/oracle/software/oracle
[oracle@racnode1 ~]$ mv linux.x64_11gR2_examples.zip /home/oracle/software/oracle
[oracle@racnode1 ~]$ cd /home/oracle/software/oracle
[oracle@racnode1 oracle]$ unzip linux.x64_11gR2_database_1of2.zip
[oracle@racnode1 oracle]$ unzip linux.x64_11gR2_database_2of2.zip
[oracle@racnode1 oracle]$ unzip linux.x64_11gR2_examples.zip
```

---

## 20. Preinstallation Tasks for Oracle Grid Infrastructure for a Cluster

*Perform the following checks on both Oracle RAC nodes in the cluster.*

This section contains any remaining preinstallation tasks for Oracle grid infrastructure that have not already been discussed. Please note that manually running the Cluster Verification Utility (CVU) before running the Oracle installer is not required. The CVU is run automatically at the end of the Oracle grid infrastructure installation as part of the Configuration Assistants process.

### Install the cvuqdisk Package for Linux

Install the operating system package `cvuqdisk` to both Oracle RAC nodes. Without `cvuqdisk`, Cluster Verification Utility cannot discover shared disks, and you receive the error message "Package `cvuqdisk` not installed" when the Cluster Verification Utility is run (either manually or at the end of the Oracle grid infrastructure installation). Use the `cvuqdisk` RPM for your hardware architecture (for example, `x86_64`, or `i386`).

The `cvuqdisk` RPM can be found on the Oracle grid infrastructure installation media in the `rpm` directory. For the purpose of this article, the Oracle grid infrastructure media was extracted to the `/home/grid/software/oracle/grid/rpm` directory on `racnode1` as the `grid` user.

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

1. Locate the `cvuqdisk` RPM package, which is in the directory `rpm` on the installation media from `racnode1`:

```
[racnode1]: /home/grid/software/oracle/grid/rpm/cvuqdisk-1.0.7-1.rpm
```

2. Copy the `cvuqdisk` package from `racnode1` to `racnode2` as the `grid` user account:

```
[racnode2]: /home/grid/software/oracle/grid/rpm/cvuqdisk-1.0.7-1.rpm
```

3. Log in as `root` on both Oracle RAC nodes:

```
[grid@racnode1 rpm]$ su
```

```
[grid@racnode2 rpm]$ su
```

4. Set the environment variable `CVUQDISK_GRP` to point to the group that will own `cvuqdisk`, which for this article is `oinstall`:

```
[root@racnode1 rpm]# CVUQDISK_GRP=oinstall; export CVUQDISK_GRP
```

```
[root@racnode2 rpm]# CVUQDISK_GRP=oinstall; export CVUQDISK_GRP
```

5. In the directory where you have saved the `cvuqdisk` RPM, use the following command to install the `cvuqdisk` package on both Oracle RAC nodes:

```
[root@racnode1 rpm]# rpm -iv cvuqdisk-1.0.7-1.rpm
Preparing packages for installation...
cvuqdisk-1.0.7-1
```

```
[root@racnode2 rpm]# rpm -iv cvuqdisk-1.0.7-1.rpm
Preparing packages for installation...
cvuqdisk-1.0.7-1
```

### Verify Oracle Clusterware Requirements with CVU - (optional)

As stated earlier in this section, running the Cluster Verification Utility before running the Oracle installer is not required. Starting with Oracle Clusterware 11g release 2, Oracle Universal Installer (OUI) detects when the minimum requirements for an installation are not met, and creates shell scripts, called fixup scripts, to finish incomplete system configuration steps. If OUI detects an incomplete task, then it generates fixup scripts (`runfixup.sh`). You can run the fixup script after you click the **Fix and Check**

**Again Button** during the Oracle grid infrastructure installation.

You also can have CVU generate fixup scripts before installation.

If you decide that you want to run the CVU, please keep in mind that it should be run as the `grid` user from the node you will be performing the Oracle installation from (`racnode1`). In addition, [SSH connectivity](#) with [user equivalence](#) must be configured for the `grid` user. If you intend to configure SSH connectivity using the OUI, the CVU utility will fail before having the opportunity to perform any of its critical checks and generate the fixup scripts:

```
Checking user equivalence...

Check: User equivalence for user "grid"
Node Name      Comment
-----
racnode2       failed
racnode1       failed
Result: PRVF-4007 : User equivalence check failed for user "grid"

ERROR:
User equivalence unavailable on all the specified nodes
Verification cannot proceed

Pre-check for cluster services setup was unsuccessful on all the nodes.
```

Once all prerequisites for running the CVU utility have been met, you can now manually check your cluster configuration before installation and generate a fixup script to make operating system changes before starting the installation.

```
[grid@racnode1 ~]$ cd /home/grid/software/oracle/grid
[grid@racnode1 grid]$ ./runcluvfy.sh stage -pre crsinst -n racnode1,racnode2 -fixup -verbose
```

Review the [CVU report](#). The only failure that should be found given the configuration described in this article is:

```
Check: Membership of user "grid" in group "dba"
Node Name      User Exists  Group Exists  User in Group  Comment
-----
racnode2       yes         yes          no             failed
racnode1       yes         yes          no             failed
Result: Membership check for user "grid" in group "dba" failed
```

The check fails because this guide creates role-allocated groups and users by using a *Job Role Separation* configuration which is not accurately recognized by the CVU. Creating a Job Role Separation configuration was described in the section [Create Job Role Separation Operating System Privileges Groups, Users, and Directories](#). The CVU fails to recognize this type of configuration and assumes the `grid` user should always be part of the `dba` group. This failed check can be safely ignored. All other checks performed by CVU should be reported as "passed" before continuing with the Oracle grid infrastructure installation.

### Verify Hardware and Operating System Setup with CVU

The next CVU check to run will verify the hardware and operating system setup. Again, run the following as the `grid` user account from `racnode1` with [user equivalence](#) configured:

```
[grid@racnode1 ~]$ cd /home/grid/software/oracle/grid  
[grid@racnode1 grid]$ ./runcluvfy.sh stage -post hwos -n racnode1,racnode2 -verbose
```

Review the [CVU report](#). All checks performed by CVU should be reported as "passed" before continuing with the Oracle grid infrastructure installation.